



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	1	

COBGDB - The GnuCOBOL TUI DEBUGGER / ANIMATOR
<i>HOW TO USE COBGDB</i>



Table of Contents

1.Introduction.....	2
1.1.Installing the debugger on Windows.....	3
1.2.Compile and Debug GnuCOBOL programs.....	4
1.3.Main Commands.....	5
2.Tutorial - Sample Debugging Session.....	6
2.1.Help Command.....	8
2.2.Run Command.....	9
2.3.Step Command.....	11
2.4.Go Command.....	13
2.5.Show Command.....	14
2.5.1.Edit subCommand.....	15
2.6.Variable Command.....	16
2.6.1.Enter subCommand.....	16
2.6.2.Edit subCommand.....	18
2.6.3.Return subCommand.....	19
2.7.Step Command.....	20
2.8.Pop-up Variable windows.....	23
2.9.File Command.....	24
2.10.Run Command.....	27
2.11.Window Size command.....	28
2.12.Quit Command.....	30
2.13.COBGDB Version.....	30
3.Document Change Log.....	31

DOCUMENT CODE	MODULE: xxxxxxxxxx	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	2	

1. Introduction

COBGDB is a TUI (Text User Interface) application, programmed in C, designed to assist in animate and debugging GnuCOBOL code **using GDB**.

The project is hosted at <https://github.com/marcsosduma/cobgdb>

***Very important: you don't need to know GDB and all its many commands
(<https://www.sourceware.org/gdb/>).***


***COBGDB has its own user interface that is very simple to use
and is responsible for interfacing the underlying GDB which is the real debug and animate engine
but operates practically in a transparent and invisible way for the user.***

The COBGDB application is based on the extension for Visual Studio Code (VSCode) created by Oleg Kunitsyn, which can be found on GitHub: <https://github.com/OlegKunitsyn/gnucobol-debug>.
warning: COBGDB is still currently under development.

***At <https://github.com/marcsosduma/cobgdb> in the Windows subdirectory,
the executable program cobgdb.exe for this operating system is available and ready to use.***

To compile COBGDB from C source code on Windows, you can use MinGW.

The Makefile is configured to generate the program **cobgdb.exe** for both Windows and Linux.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	3	

1.1. Installing the debugger on Windows

On Windows, just download **cobgdb.exe** from following folder:
<https://github.com/marcsosduma/cobgdb/tree/main/windows>.

As an example you can put **cobgdb.exe** into the "bin" folder of your GnuCOBOL installation (the same folder where the GnuCOBOL compiler **cobc.exe** is located)

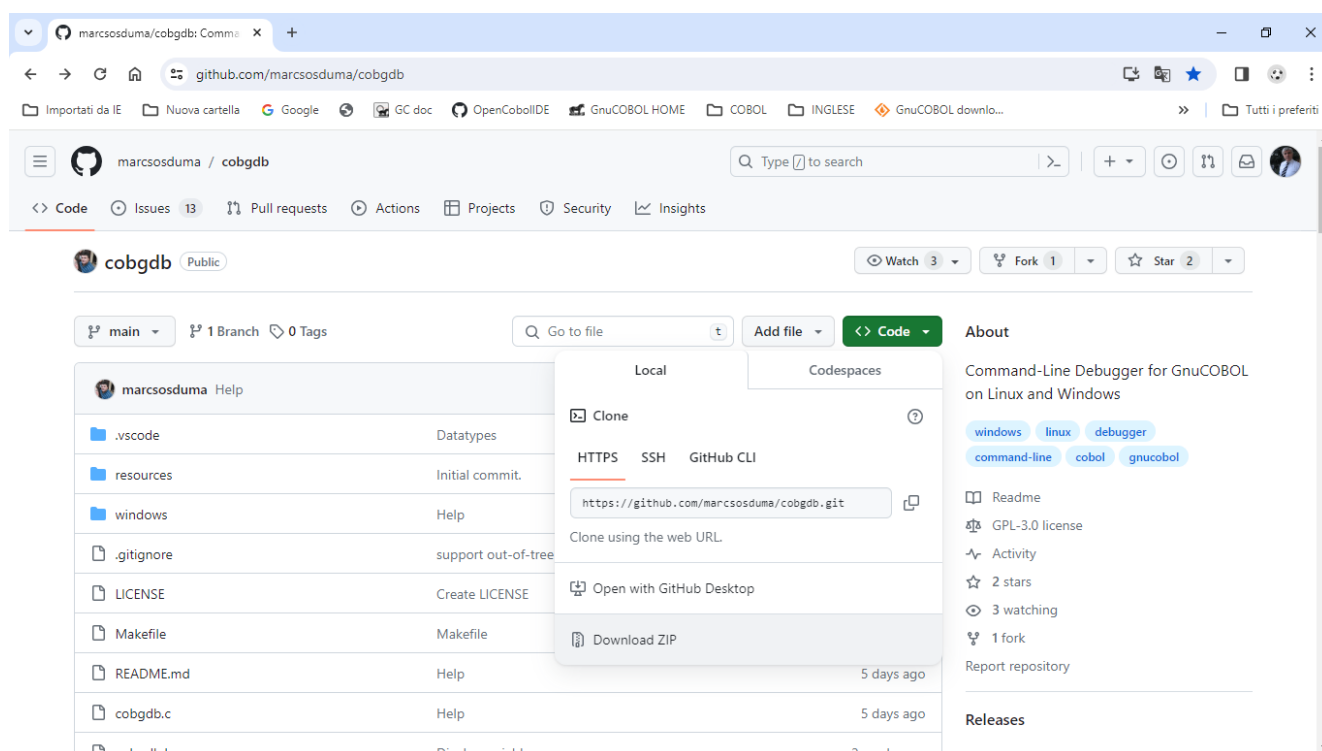
or


first install MinGW (Minimalist GNU for Windows).

Then execute the make ('mingw32-make' for Windows) command to compile the code from C source.

Note: if you have security problems downloading an .exe file then you can try download the entire repository with the following github button **<> Code v** --> **Download ZIP**

Then unzip the file and copy cobgdb.exe to the "bin" folder of your GnuCOBOL installation (the same folder where the GnuCOBOL compiler cobc.exe is located)



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	4	

1.2. Compile and Debug GnuCOBOL programs

Compile and run a debugging session of the sample program using the following command:

cobgdb customer.cob -x -lpdcurses

Source code of customer.cob used also for following tutorial is at:

<https://github.com/marcsosduma/cobgdb/tree/main/windows>

Note: '-lpdcurses' is an instance of an argument that can be indirectly passed to 'cobc' by 'cobgdb,' even if it is not used by 'cobgdb' itself.

or, other example for cobc parameters , use : **cobgdb customer.cob -x -Tcustomer.txt** .(-T creates a compilation list output into **customer.txt** file)

COBGDB takes one or more programs with COB or CBL extension as parameters and runs the GnuCOBOL compiler with the following format:

cobc -g -fsource-location -ftraceall -v -free -O0 -x prog.cob prog2.cob ...

To debug multiple programs, use COBGDB with the following syntax:

cobgdb prog.cob subprog1.cob subprog2.cob ...


This will create a single prog.exe executable.

You can run GDB/GDBSERVER remotely using the "A" (Attach) key.

COBGDB will prompt you to provide the server and port in the format server: port or the PID of the application.


Example:

- localhost: 5555
- 9112

DOCUMENT CODE	MODULE: xxxxxxxxxx	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	5	

1.3. Main Commands

Cmd		Description	
?	Help	Show the HELP window and text	
R	Run	Runs the program <u>from the first Cobol statement</u> until a breakpoint is encountered. Always use this command to start a debugging session.	
B	Breakpoint	Toggle (Set or Unset) a breakpoint at a specific line of the Procedure Division code.	
C	Cursor	Runs the program until it reaches the selected line.	
N	Next	Runs the program until the next line but does not enter a subroutine executed by CALL or PERFORM.	
S	Step	Runs the program until the next line. If needed it enter a subroutine executed by CALL or PERFORM.	
G	Go	Continues the program execution until it encounters a stopping point: a breakpoint, end of the program, or the return from a subroutine (PERFORM / CALL).	
J	Jump	Ask for a line number and Runs the program until it reaches that line.	
V	Variables	Displays a window with a list of all variables for the running program. From this window you can also change the value of one or more variables.	
H	sHow	Displays a window with a list of variables from the selected line. From this window you can also edit / change the value of variables from the selected line.	
F	File	When cobgdb is executed with more than one program, allows selecting one of those source file to manage debugging commands.	
A	Attach	Attach to GDBSERVER or Application PID.	
W	Window size	Switch between two window size of the debugger : 24 x 80 or 34 x 132.	
Q	Quit	Quits (ends) the debugging session and the program (or programs).	

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	6	

2. Tutorial - Sample Debugging Session

Following tutorial is on a Windows 10 platform using following version of GnuCOBOL:

```
cobc (GnuCOBOL) 3.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Keisuke Nishida, Roger While, Ron Norman, Simon Sobisch, Edward Hart
Built Jul 28 2023 16:07:38
Packaged Jul 28 2023 16:58:47 UTC
C version (MinGW) "13.1.0"
```

Downloaded from <https://www.arnoldtrembley.com/GnuCOBOL.htm>


===== Version 3.2 =====

GnuCOBOL 3.2 (28Jul2023) MSYS2 64-bit [GC32M-BDB-x64.7z](#) -- MSYS2 64-bit GnuCOBOL 3.2

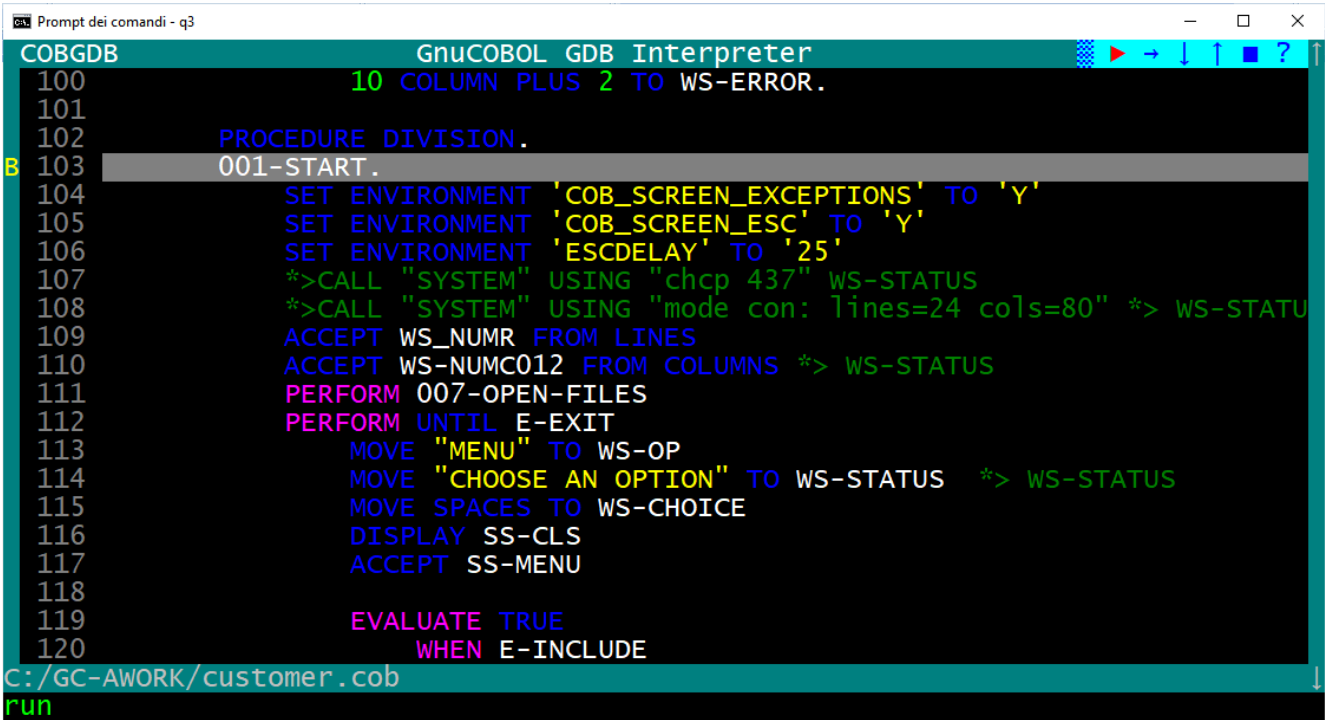
Final release **with full debugging support**. (95.4 Megabytes).

GnuCOBOL 3.2 (28Jul2023) MSYS2 32-bit [GC32M-BDB-x32.7z](#) -- MSYS2 32-bit GnuCOBOL 3.2

Final release **with full debugging support**. (96.1 Megabytes).

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	7	

After executing **cobgdb customer.cob -x -lpdcurses** the application automatically insert a Breakpoint at first executable program statement of PROCEDURE DIVISION (see the **B** symbol at left of line 103 in this sample) and displays following screen:



```

COBGDB                               GnuCOBOL GDB Interpreter
100          10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103  B  001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116              DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
run

```


You can scroll the source code window with cursor keys UP and DOWN, PG UP and PG DOWN or with mouse wheel or with mouse left click on the right scroll bar. Use cursor RIGHT and cursor LEFT to scroll horizontally,

In the upper right window corner there is a "button bar" where you can find some buttons (symbols):


>
→
↓
↑
■
?

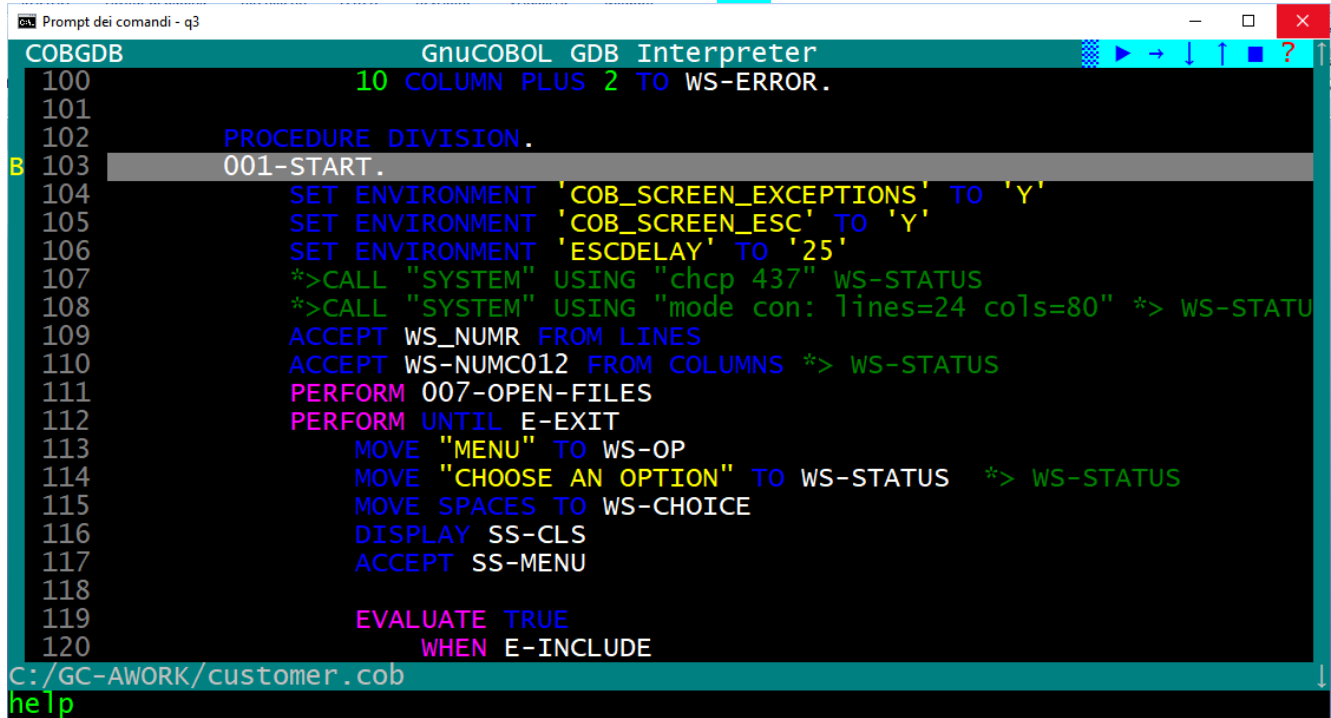
>	=	Run command
→	=	Next command
↓	=	Step command
↑	=	Go command
■	=	Quit command
?	=	Help command

when you hover over one of these symbol, you get the corresponding command description (like a tooltip) displayed at the bottom left of the screen.

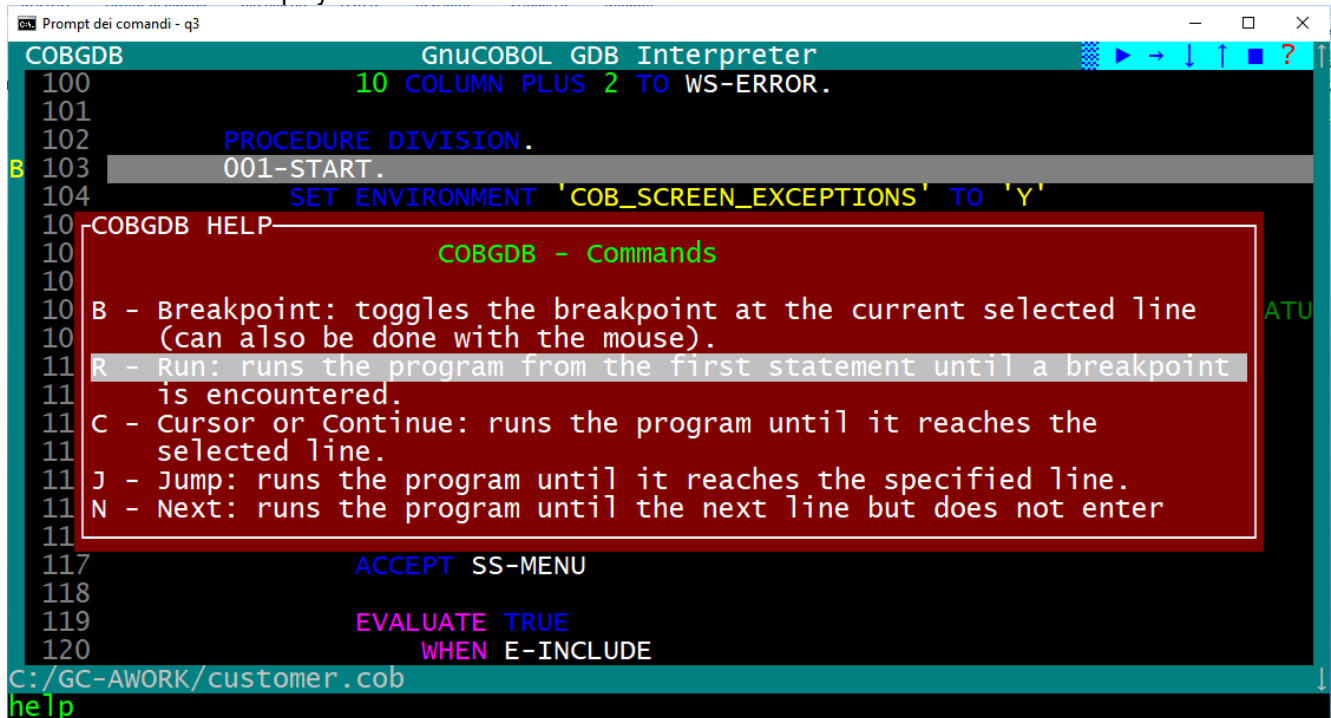
DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	8	

2.1. Help Command

Type ? (key) HELP command or left click with mouse on the  button:




the HELP window is displayed



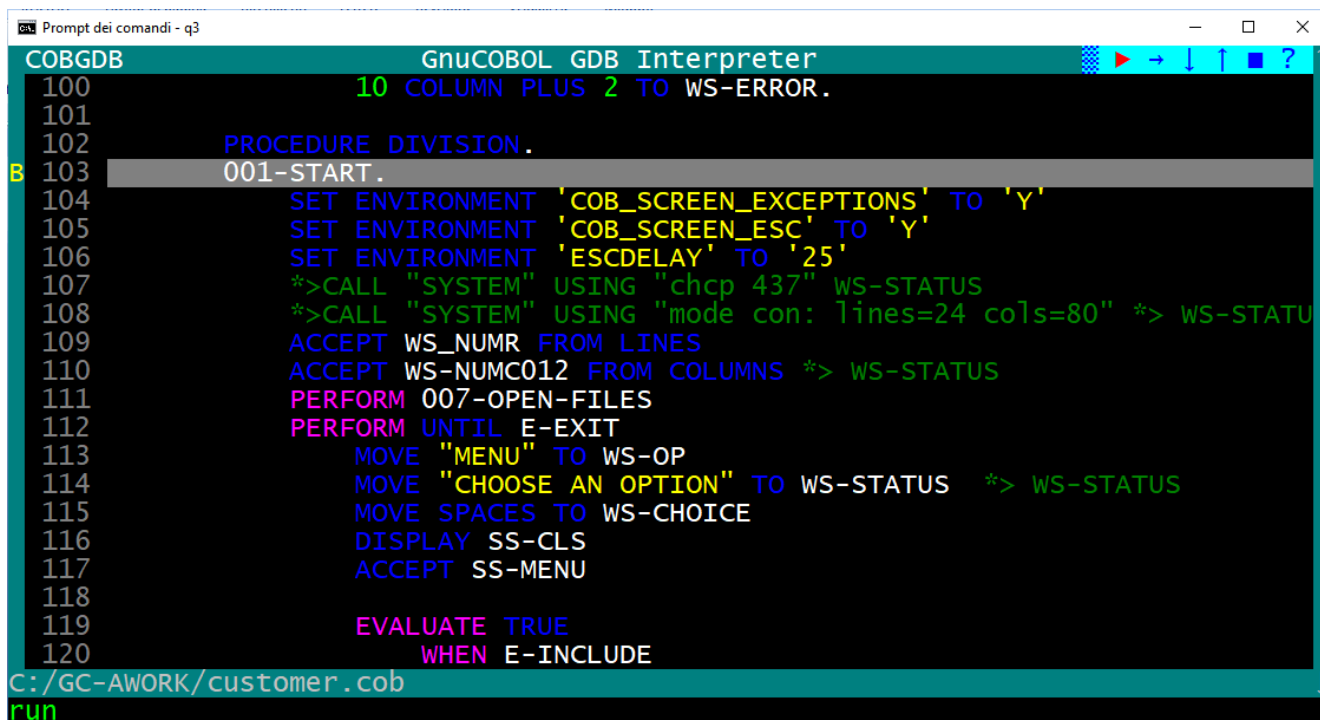
scroll the Help window with cursor keys UP and DOWN or mouse wheel.

Use ESC or Enter or left click to exit from this HELP window and return to debugging session.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	9	

2.2. Run Command

To start executing the program and the debugging session from first program statement you always must use the "R" command (key) or left click with mouse on the  **Run** button




```

COBGDB GnuCOBOL GDB Interpreter
100      10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116              DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
run

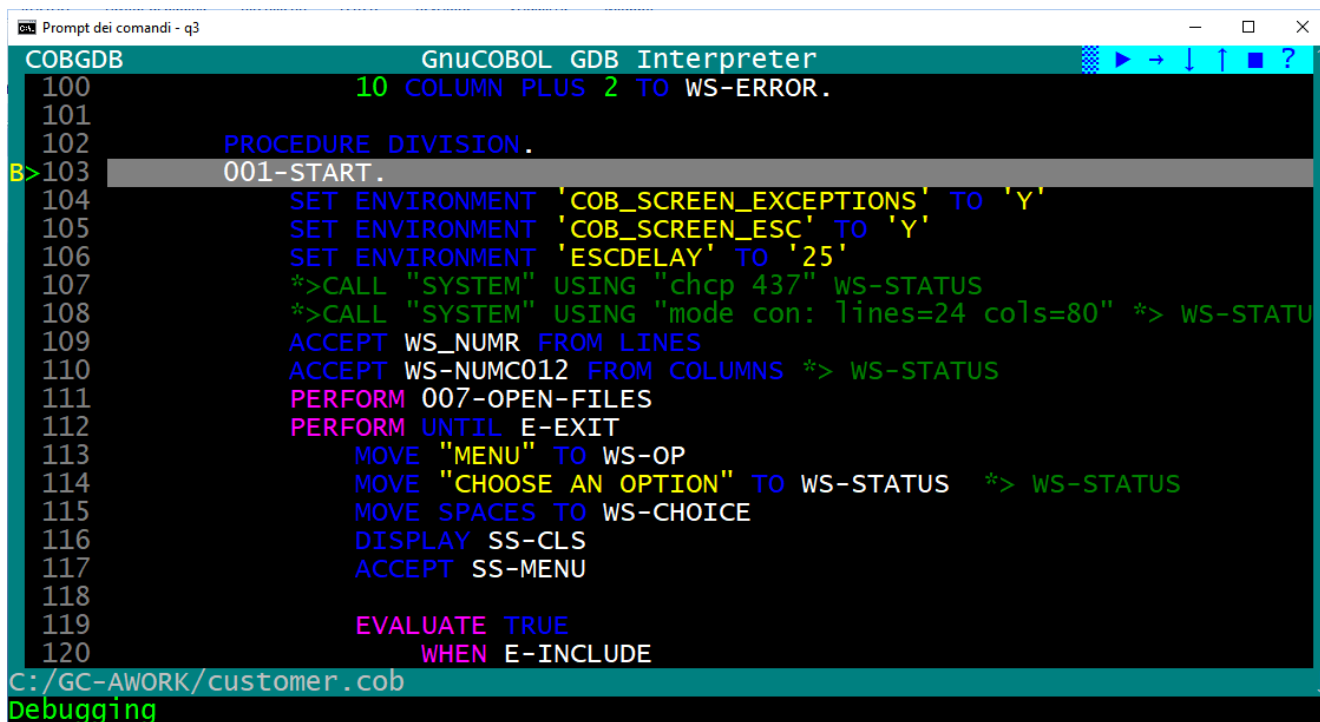
```

cobgdb opens the program terminal window (the application will run in this separate window.).



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	10	

go back to the COBGDB screen, and you see a  green symbol on the left of statement where initial Breakpoint is present (in our example is at line 103 **B>103**):




```

COBGDB GnuCOBOL GDB Interpreter
100      10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
B>103    001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116              DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Debugging


```

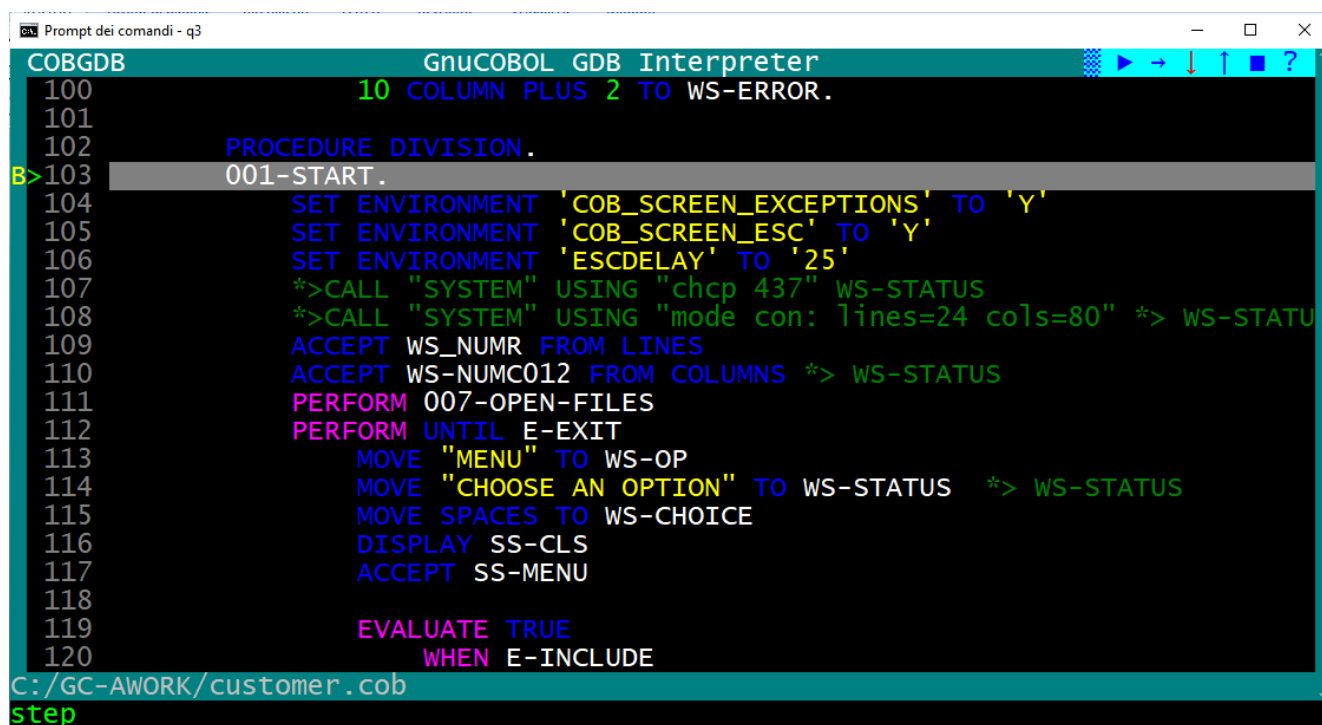
From that moment on, you can use all the commands (keys) or corresponding buttons to "animate" and debug the application, example: **"S"** (Step), **"N"** (Next), **"G"** (Go) and so on.

During the source code animation, the debugger automatically shows some pop-up windows with variables content from the line in execution.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	11	

2.3. Step Command

Proceed with **S** (Step) command or left click with mouse the  button:.

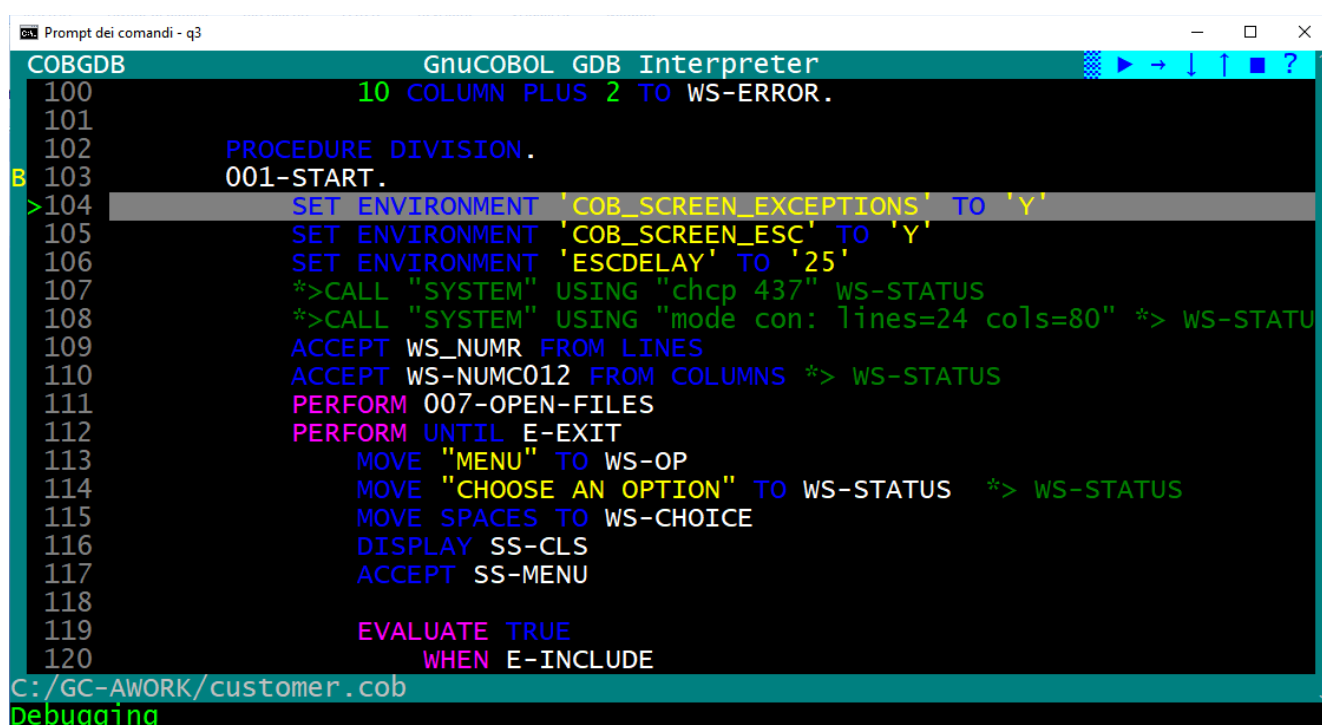


```

COBGDB                                GnuCOBOL GDB Interpreter
100                                10 COLUMN PLUS 2 TO WS-ERROR.
101
102                                PROCEDURE DIVISION.
B>103                                001-START.
104                                SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105                                SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106                                SET ENVIRONMENT 'ESCDELAY' TO '25'
107                                *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108                                *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109                                ACCEPT WS_NUMR FROM LINES
110                                ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
111                                PERFORM 007-OPEN-FILES
112                                PERFORM UNTIL E-EXIT
113                                    MOVE "MENU" TO WS-OP
114                                    MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115                                    MOVE SPACES TO WS-CHOICE
116                                    DISPLAY SS-CLS
117                                    ACCEPT SS-MENU
118
119                                    EVALUATE TRUE
120                                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
step

```


the **>104** green symbol now is on the following line 104:



```

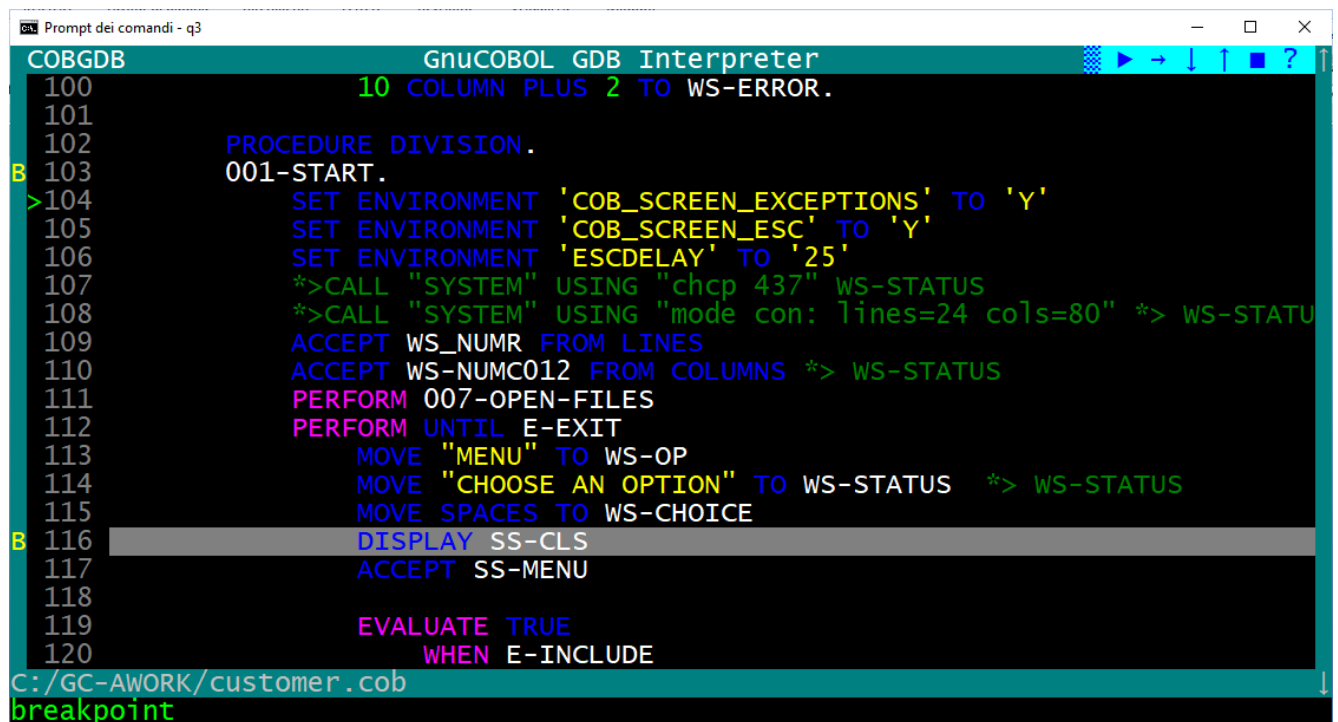
COBGDB                                GnuCOBOL GDB Interpreter
100                                10 COLUMN PLUS 2 TO WS-ERROR.
101
102                                PROCEDURE DIVISION.
B>103                                001-START.
>104                                SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105                                SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106                                SET ENVIRONMENT 'ESCDELAY' TO '25'
107                                *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108                                *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109                                ACCEPT WS_NUMR FROM LINES
110                                ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
111                                PERFORM 007-OPEN-FILES
112                                PERFORM UNTIL E-EXIT
113                                    MOVE "MENU" TO WS-OP
114                                    MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115                                    MOVE SPACES TO WS-CHOICE
116                                    DISPLAY SS-CLS
117                                    ACCEPT SS-MENU
118
119                                    EVALUATE TRUE
120                                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Debugging

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	12	

Now you can proceed with S command or N command or as an example:


- Scroll with cursor down to select line 116 of Procedure Division and type "B" (to set a Breakpoint), (you also can simply click with mouse left button on the 116 row number)
- The application displays a "B" on the left of the line (type B again - or re-click - when you want to delete the Breakpoint, not do that at this moment)





```

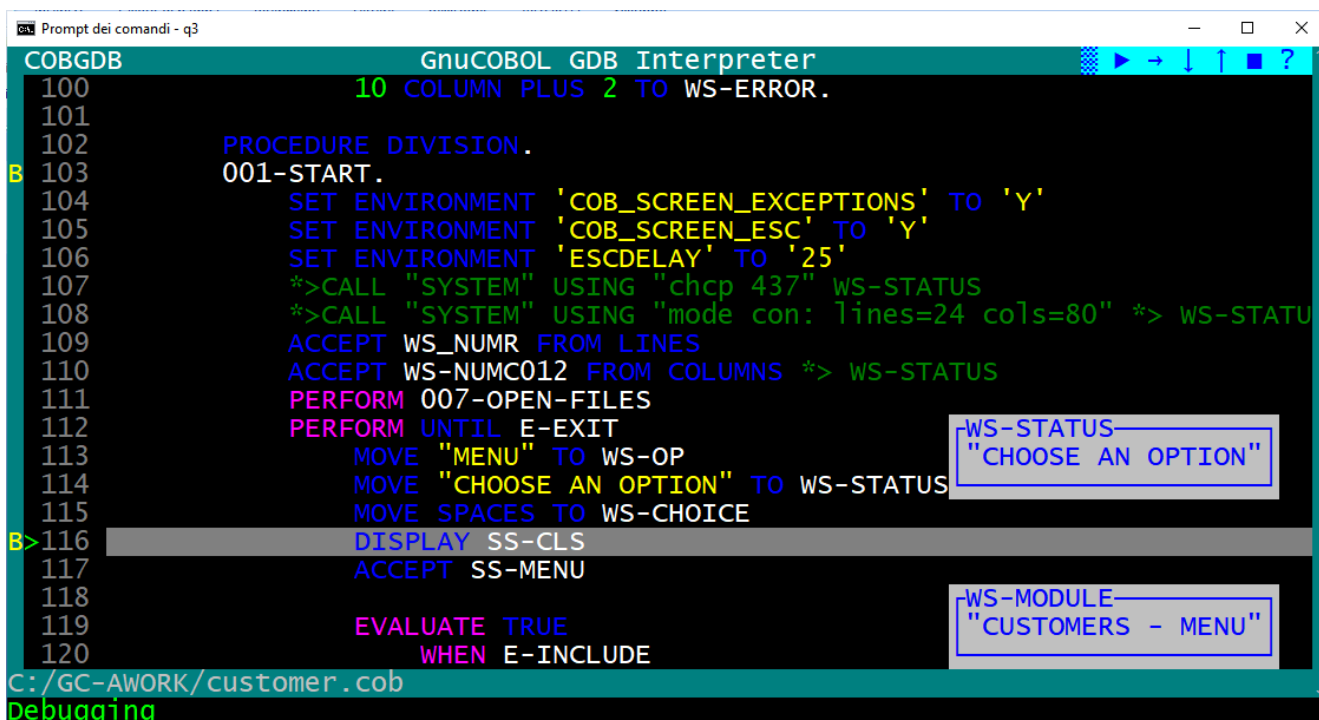
COBGDB GnuCOBOL GDB Interpreter
100 10 COLUMN PLUS 2 TO WS-ERROR.
101
102 PROCEDURE DIVISION.
103 001-START.
104 SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105 SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106 SET ENVIRONMENT 'ESCDELAY' TO '25'
107 *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108 *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109 ACCEPT WS_NUMR FROM LINES
110 ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111 PERFORM 007-OPEN-FILES
112 PERFORM UNTIL E-EXIT
113     MOVE "MENU" TO WS-OP
114     MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115     MOVE SPACES TO WS-CHOICE
116 B  DISPLAY SS-CLS
117     ACCEPT SS-MENU
118
119     EVALUATE TRUE
120         WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
breakpoint

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	13	

2.4. Go Command

Type **G** (Go) or left click with mouse the  button to execute the program until a B Breakpoint is detected: the system reach the second breakpoint at line 116 and displays a green  symbol to the left of the line to be executed, see following screen:




```

COBGDB GnuCOBOL GDB Interpreter
100      10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116      B> 116      DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
  
```

C:/GC-AWORK/customer.cob
Debugging

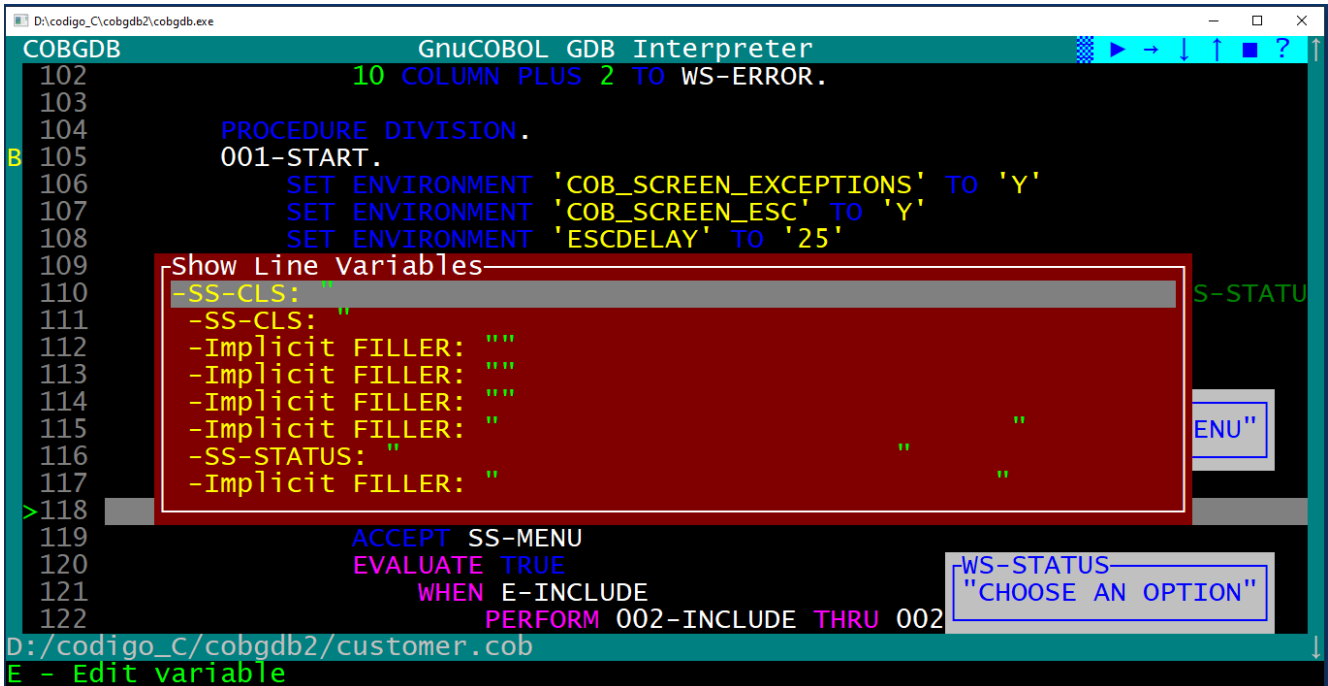
WS-STATUS
"CHOOSE AN OPTION"

WS-MODULE
"CUSTOMERS - MENU"

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	14	

2.5. Show Command

typing the 'H' command (key) allows you to view the variables on the highlighted line.

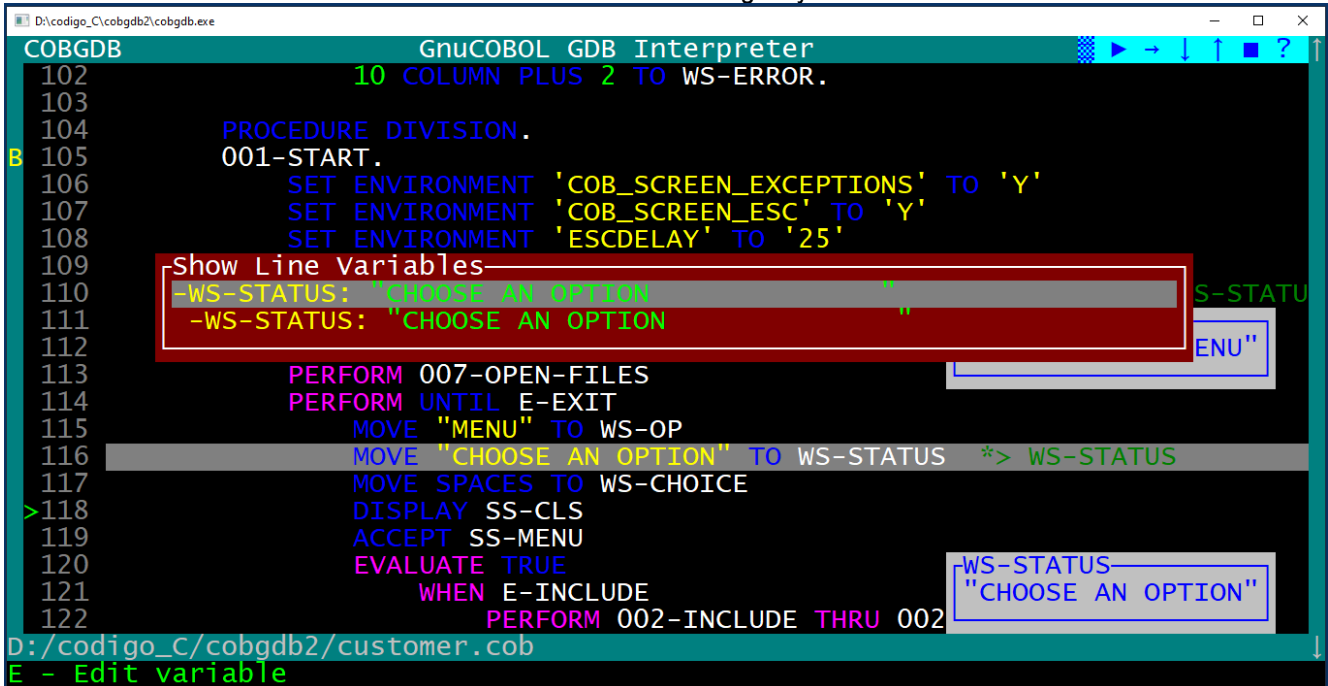


```

COBGDB GnuCOBOL GDB Interpreter
102      10 COLUMN PLUS 2 TO WS-ERROR.
103
104      PROCEDURE DIVISION.
105      001-START.
106          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
107          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
108          SET ENVIRONMENT 'ESCDELAY' TO '25'
109
110      Show Line Variables
111      -SS-CLS: "
112      -Implicit FILLER: ""
113      -Implicit FILLER: ""
114      -Implicit FILLER: ""
115      -Implicit FILLER: ""
116      -SS-STATUS: "CHOOSE AN OPTION"
117      -Implicit FILLER: ""
118
119      ACCEPT SS-MENU
120      EVALUATE TRUE
121      WHEN E-INCLUDE
122      PERFORM 002-INCLUDE THRU 002
WS-STATUS: "CHOOSE AN OPTION"
D:/codigo_C/cobgdb2/customer.cob
E - Edit variable

```

Display the content of variables also clicking right mouse button on a source line, example click right mouse button on line 116 will execute the H command on that line and give you:




```

COBGDB GnuCOBOL GDB Interpreter
102      10 COLUMN PLUS 2 TO WS-ERROR.
103
104      PROCEDURE DIVISION.
105      001-START.
106          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
107          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
108          SET ENVIRONMENT 'ESCDELAY' TO '25'
109
110      Show Line Variables
111      -WS-STATUS: "CHOOSE AN OPTION"
112      -WS-STATUS: "CHOOSE AN OPTION"
113
114      PERFORM 007-OPEN-FILES
115      PERFORM UNTIL E-EXIT
116      MOVE "MENU" TO WS-OP
117      MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
118      MOVE SPACES TO WS-CHOICE
119      DISPLAY SS-CLS
120      ACCEPT SS-MENU
121      EVALUATE TRUE
122      WHEN E-INCLUDE
123      PERFORM 002-INCLUDE THRU 002
WS-STATUS: "CHOOSE AN OPTION"
D:/codigo_C/cobgdb2/customer.cob
E - Edit variable

```

Now at lower left line on screen is a message: E = Edit (change) the variable value.

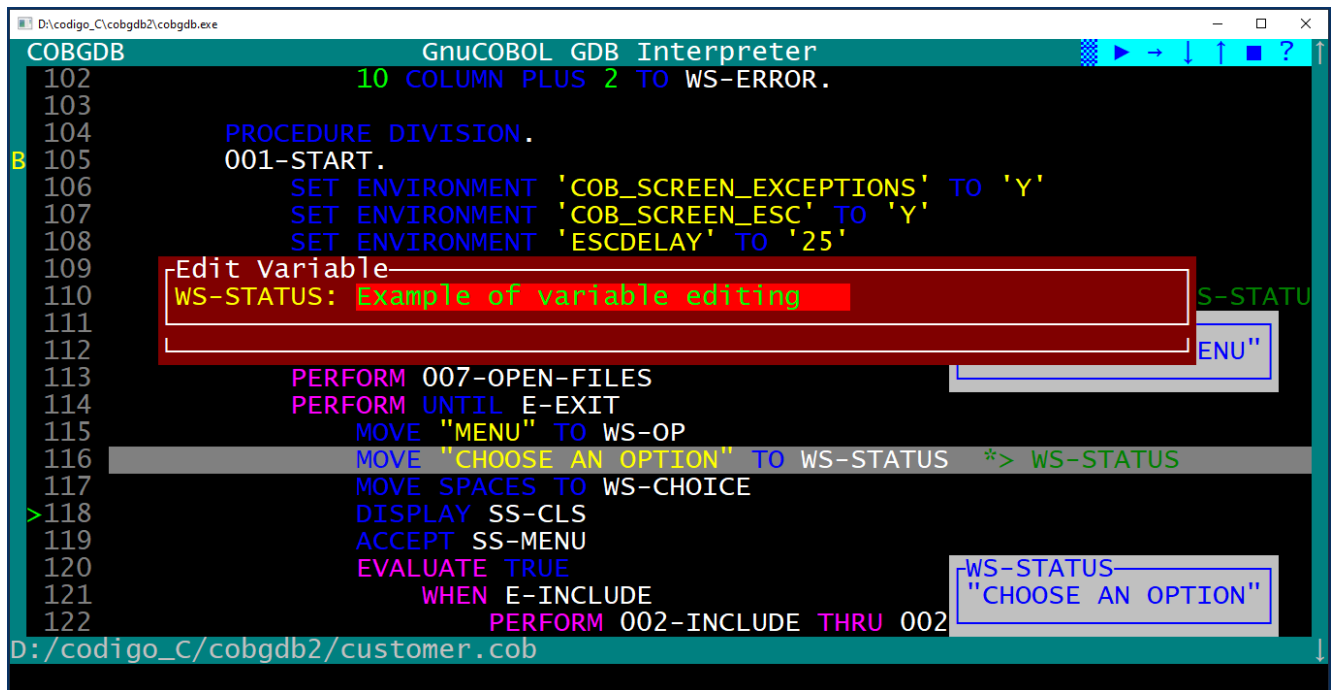
DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	15	

2.5.1.Edit subCommand

Scroll with cursor key UP and DOWN (or with mouse wheel). to select one of the variable.

The key **E** can be used to edit the content of the highlighted variable.

Change the value and type Enter to confirm changes or use ESC to exit without changes :

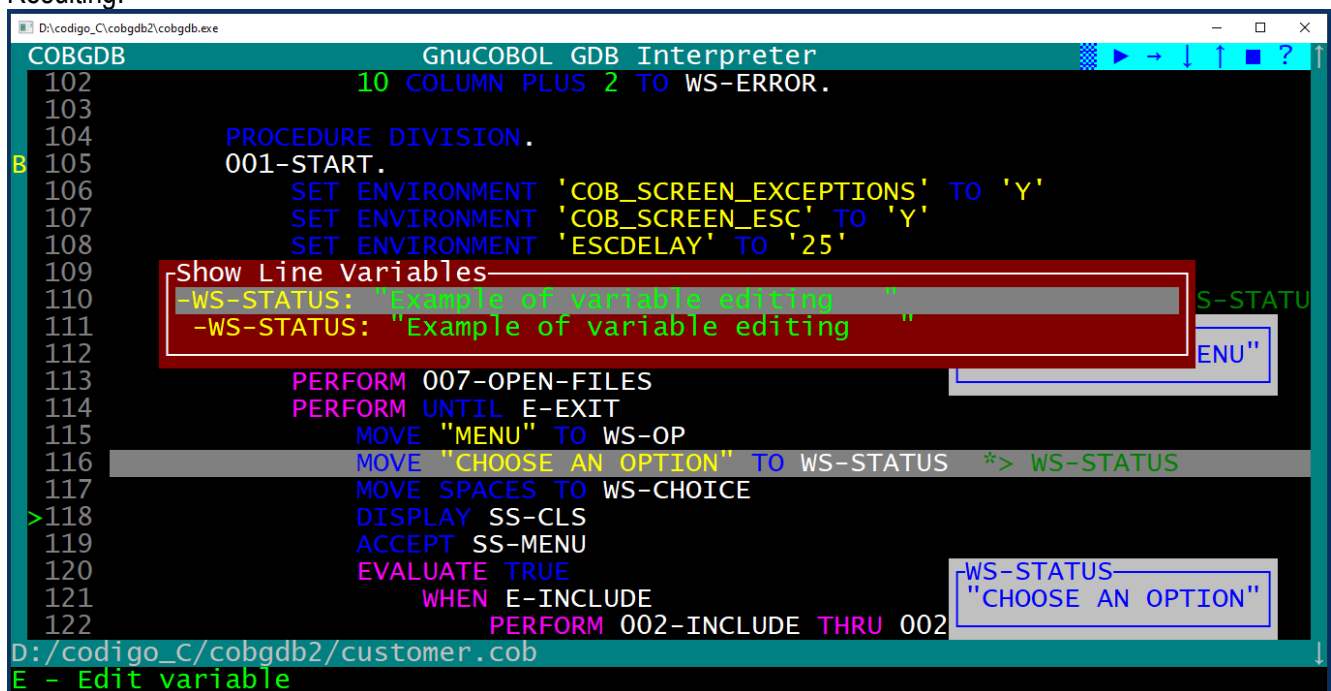


```

COBGDB GnuCOBOL GDB Interpreter
102      10 COLUMN PLUS 2 TO WS-ERROR.
103
104      PROCEDURE DIVISION.
105      001-START.
106          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
107          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
108          SET ENVIRONMENT 'ESCDELAY' TO '25'
109
110      Edit Variable
111      WS-STATUS: Example of variable editing
112
113          PERFORM 007-OPEN-FILES
114          PERFORM UNTIL E-EXIT
115              MOVE "MENU" TO WS-OP
116              MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
117              MOVE SPACES TO WS-CHOICE
118              DISPLAY SS-CLS
119              ACCEPT SS-MENU
120              EVALUATE TRUE
121                  WHEN E-INCLUDE
122                      PERFORM 002-INCLUDE THRU 002
123
124      WS-STATUS: "CHOOSE AN OPTION"
125
D:/codigo_c/cobgdb2/customer.cob

```


Resulting:



```

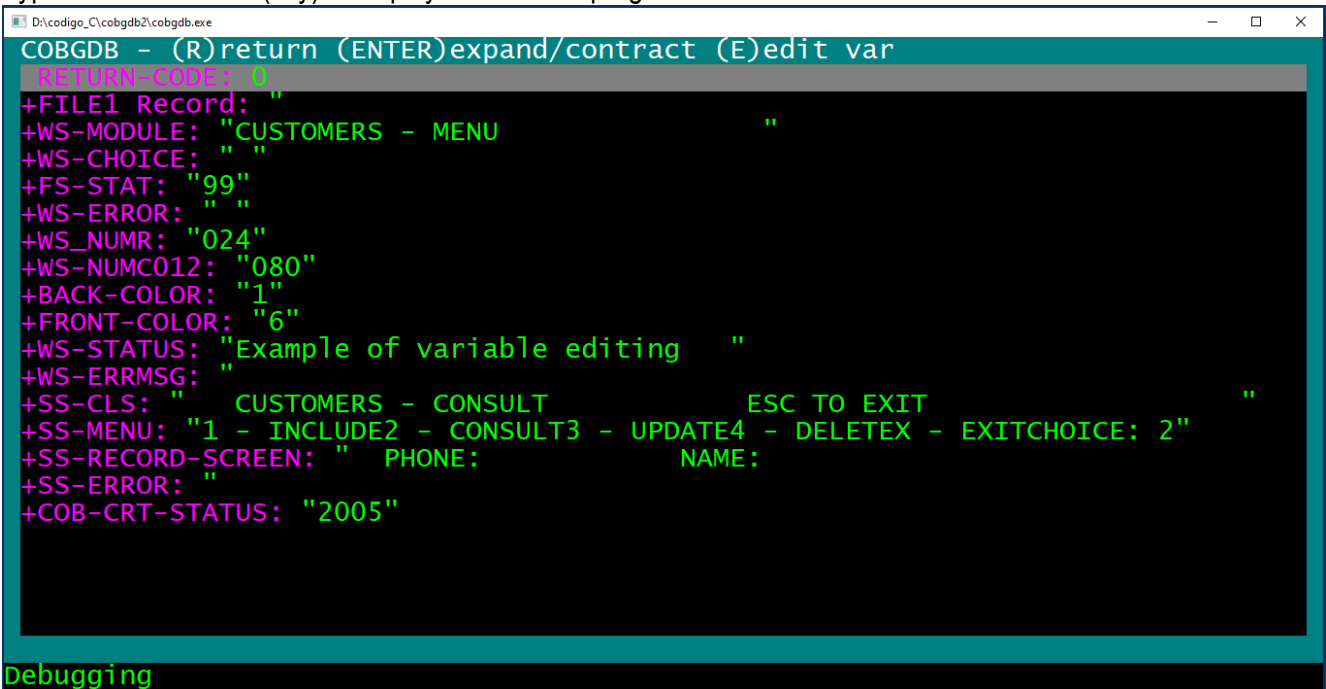
COBGDB GnuCOBOL GDB Interpreter
102      10 COLUMN PLUS 2 TO WS-ERROR.
103
104      PROCEDURE DIVISION.
105      001-START.
106          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
107          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
108          SET ENVIRONMENT 'ESCDELAY' TO '25'
109
110      Show Line Variables
111      -WS-STATUS: "Example of variable editing"
112      -WS-STATUS: "Example of variable editing"
113
114          PERFORM 007-OPEN-FILES
115          PERFORM UNTIL E-EXIT
116              MOVE "MENU" TO WS-OP
117              MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
118              MOVE SPACES TO WS-CHOICE
119              DISPLAY SS-CLS
120              ACCEPT SS-MENU
121              EVALUATE TRUE
122                  WHEN E-INCLUDE
123                      PERFORM 002-INCLUDE THRU 002
124
125      WS-STATUS: "CHOOSE AN OPTION"
126
D:/codigo_c/cobgdb2/customer.cob
E - Edit variable

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	16	

2.6. Variable Command

Type the **V** command (key) to display the list of all program variables:



```

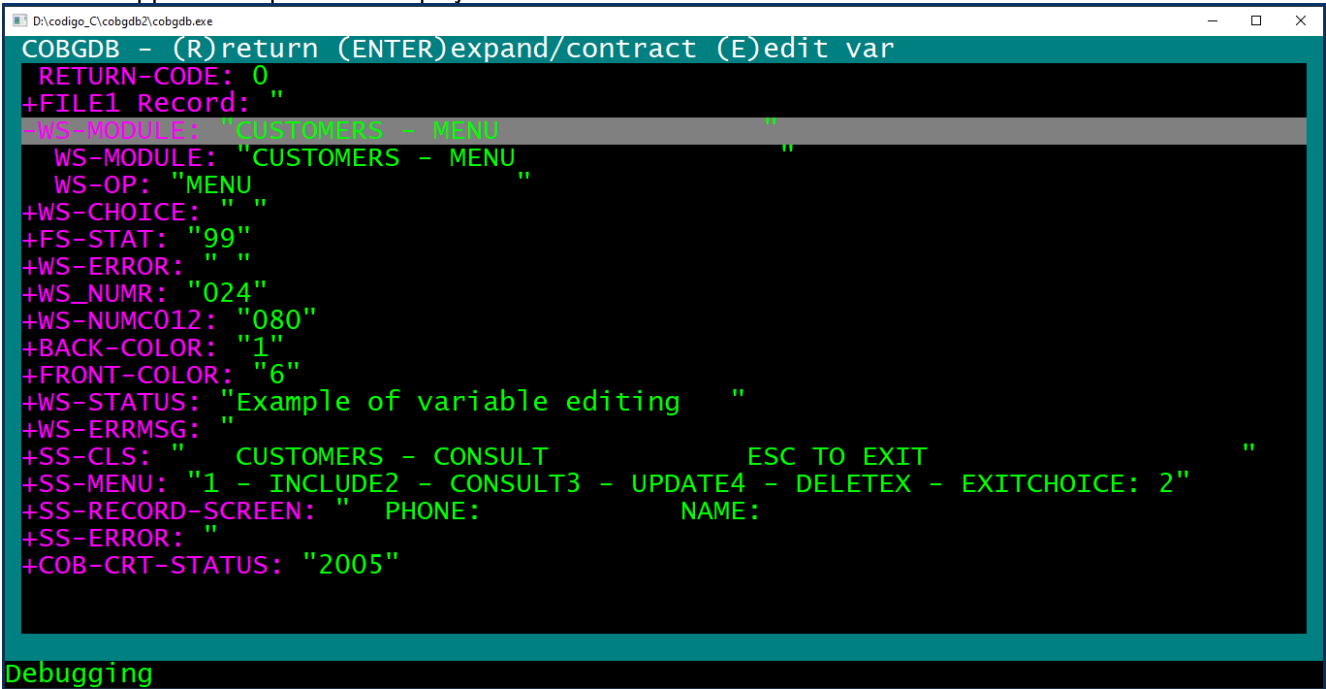
D:\codigo_C\cobgdb2\cobgdb.exe
COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
+WS-MODULE: "CUSTOMERS - MENU"
+WS-CHOICE: " "
+FS-STAT: "99"
+WS-ERROR: " "
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing"
+WS-ERRMSG: "
+SS-CLS: "CUSTOMERS - CONSULT" ESC TO EXIT"
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "PHONE: NAME:"
+SS-ERROR: "
+COB-CRT-STATUS: "2005"

Debugging

```

2.6.1. Enter subCommand

Scroll with cursor key UP and DOWN (or with mouse wheel) in this list to the variable WS-MODULE and type Enter: the application opens and displays its subfields WS-MODULE and WS-OP




```

D:\codigo_C\cobgdb2\cobgdb.exe
COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
+WS-MODULE: "CUSTOMERS - MENU"
  WS-MODULE: "CUSTOMERS - MENU"
  WS-OP: "MENU"
+WS-CHOICE: " "
+FS-STAT: "99"
+WS-ERROR: " "
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing"
+WS-ERRMSG: "
+SS-CLS: "CUSTOMERS - CONSULT" ESC TO EXIT"
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "PHONE: NAME:"
+SS-ERROR: "
+COB-CRT-STATUS: "2005"

Debugging

```


DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	17	


select WS-MODULE subfield

```

D:\codigo_C\cobgdb2\cobgdb.exe
COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
-WS-MODULE: "CUSTOMERS - MENU"
WS-MODULE: "CUSTOMERS - MENU"
WS-OP: "MENU"
+WS-CHOICE: " "
+FS-STAT: "99"
+WS-ERROR: " "
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing "
+WS-ERRMSG: "
+SS-CLS: " CUSTOMERS - CONSULT ESC TO EXIT "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: " PHONE: NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "2005"

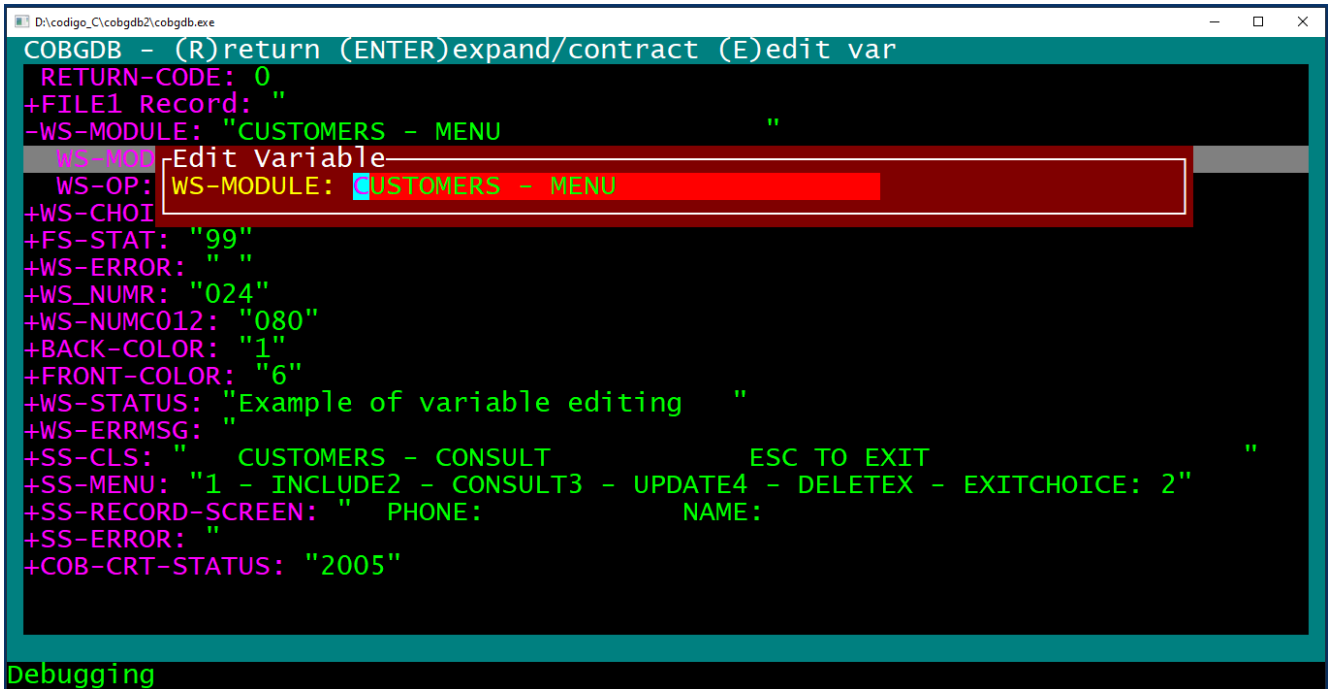
Debugging

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	18	

2.6.2.Edit subCommand

Now you can select WS-MODULE subfield whit cursor DOWN or mouse wheel and type "E" (Edit)
COBGDB shows a Edit Variable window:



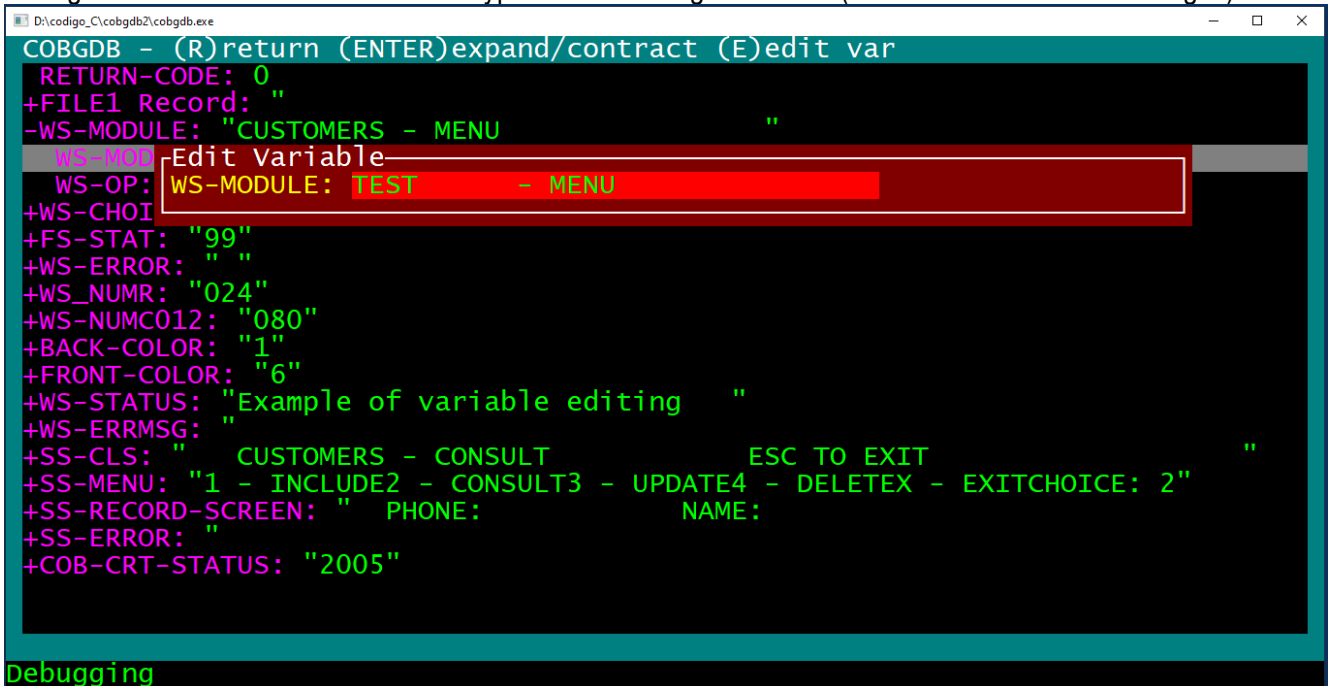
```

COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
-WS-MODULE: "CUSTOMERS - MENU"
WS-MOD Edit Variable
WS-OP: WS-MODULE: CUSTOMERS - MENU
+WS-CHOI
+FS-STAT: "99"
+WS-ERROR: ""
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing"
+WS-ERRMSG: ""
+SS-CLS: "CUSTOMERS - CONSULT ESC TO EXIT"
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "PHONE: NAME:"
+SS-ERROR: ""
+COB-CRT-STATUS: "2005"

Debugging

```

Change "CUSTOMERS" to "TEST" and type Enter to change the value (use ESC to exit without changes) :




```

COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
-WS-MODULE: "CUSTOMERS - MENU"
WS-MOD Edit Variable
WS-OP: WS-MODULE: TEST - MENU
+WS-CHOI
+FS-STAT: "99"
+WS-ERROR: ""
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing"
+WS-ERRMSG: ""
+SS-CLS: "CUSTOMERS - CONSULT ESC TO EXIT"
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "PHONE: NAME:"
+SS-ERROR: ""
+COB-CRT-STATUS: "2005"

Debugging

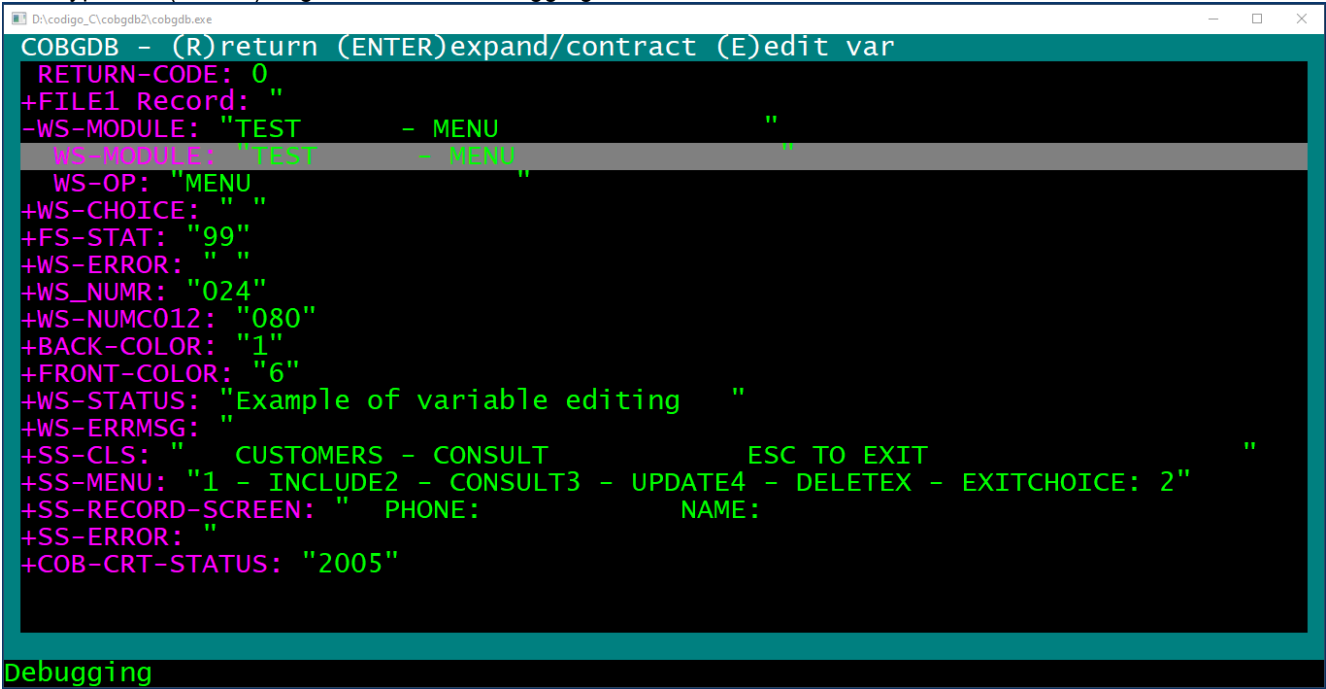
```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	19	

2.6.3. Return subCommand

WS-Module has new value.

Now type "R" (Return) to go back to the debugging session:




The screenshot shows a terminal window titled "COBGDB - (R)return (ENTER)expand/contract (E)edit var". The output of the return command is as follows:

```

RETURN-CODE: 0
+FILE1 Record: "
-WS-MODULE: "TEST      - MENU      "
WS-MODULE: "TEST      - MENU      "
WS-OP: "MENU"
+WS-CHOICE: " "
+FS-STAT: "99"
+WS-ERROR: " "
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing  "
+WS-ERRMSG: "
+SS-CLS: "      CUSTOMERS - CONSULT      ESC TO EXIT      "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "  PHONE:      NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "2005"

```

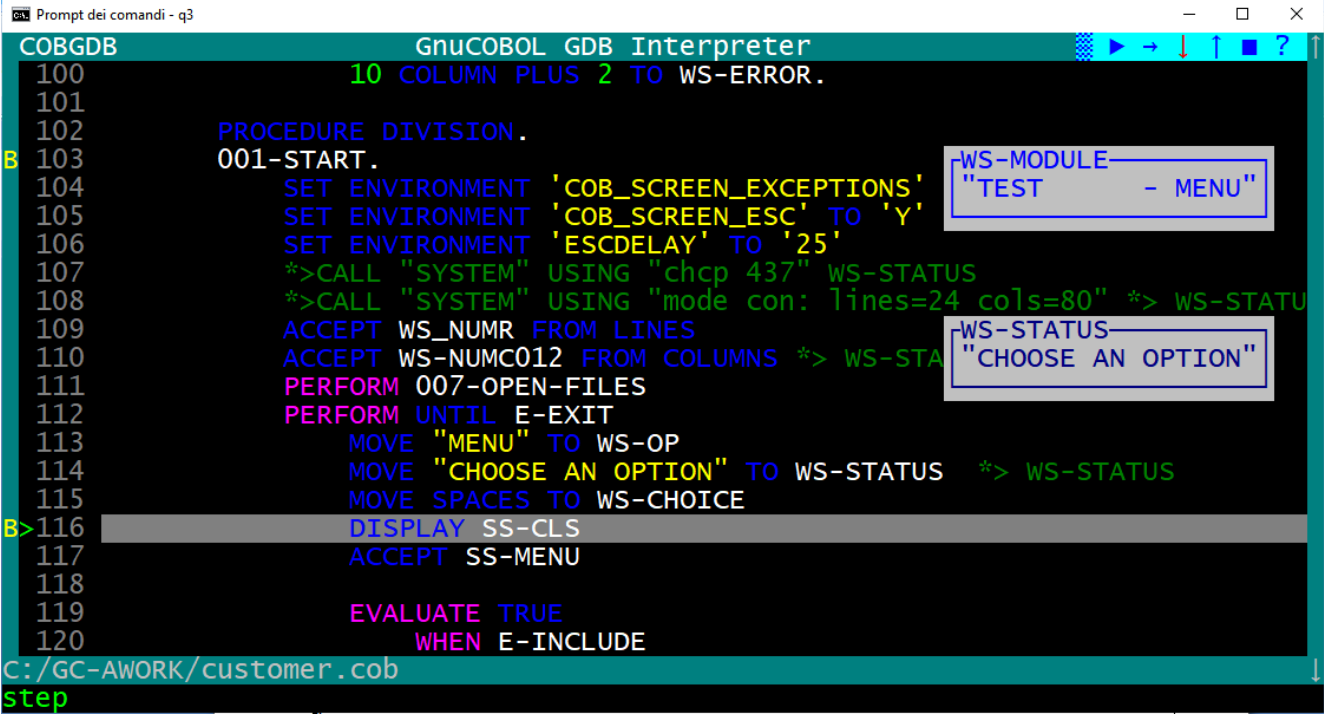
At the bottom of the window, the word "Debugging" is displayed in green text.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	20	

2.7. Step Command

now you are back in the main debugging window:

Type **S** (Step) command or left click with mouse the  button to execute the DISPLAY statement at line 116



```

COBGDB GnuCOBOL GDB Interpreter
100      10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116      B> DISPLAY SS-CLS
117          ACCEPT SS-MENU
118
119          EVALUATE TRUE
120              WHEN E-INCLUDE

```

C:/GC-AWORK/customer.cob
step

in the other application window you can see the result of DISPLAY statement




```

C:\GC-AWORK\customer.exe
TEST - MENU

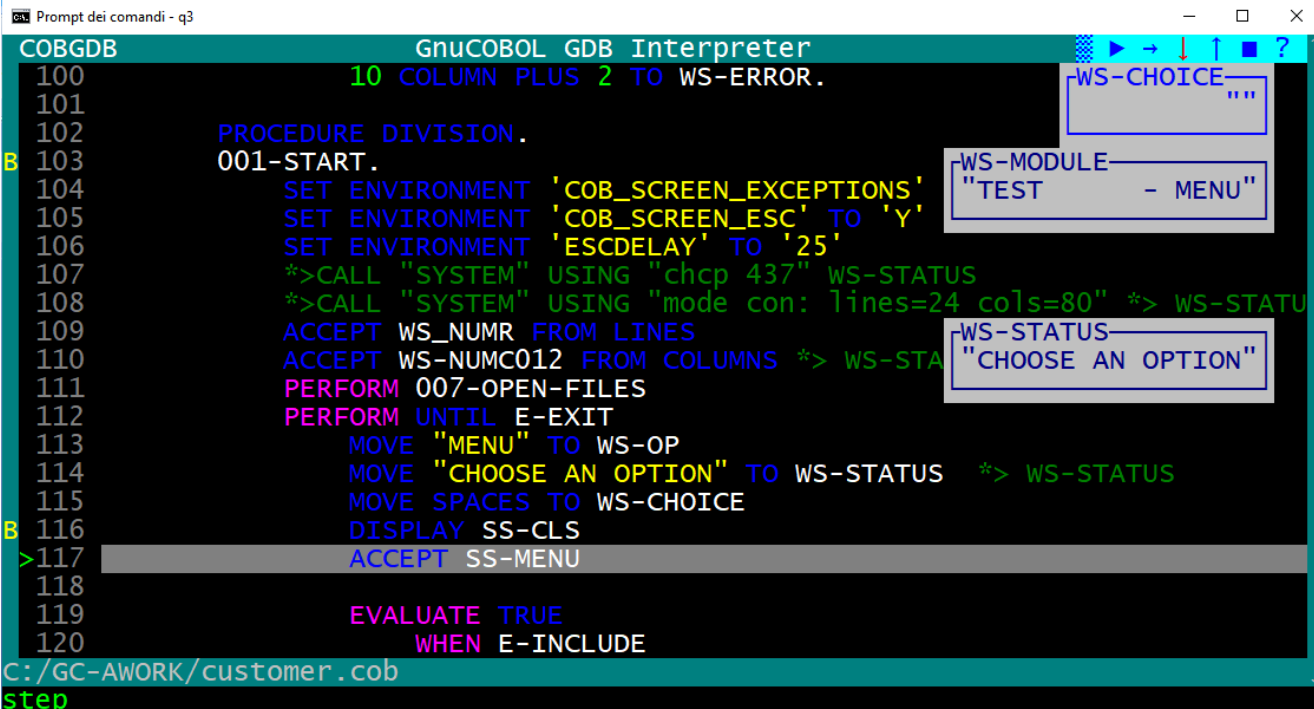
CHOOSE AN OPTION

```

go back to debugging window and now the ACCEPT statement will be executed with S command

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	21	

Type **S** (Step) command or leftclick with mouse the  button again to execute ACCEPT statement at line 117:



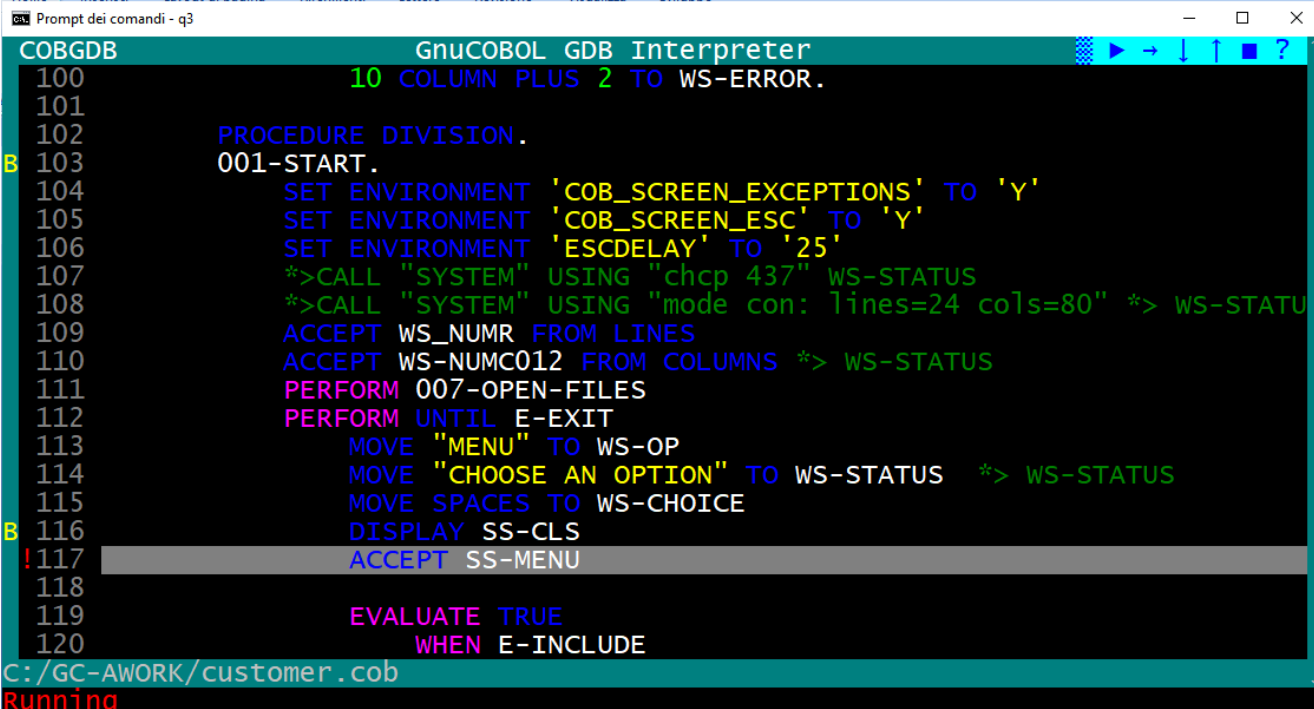
```

COBGDB GnuCOBOL GDB Interpreter
100 10 COLUMN PLUS 2 TO WS-ERROR.
101
102 PROCEDURE DIVISION.
103 001-START.
104 SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS'
105 SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106 SET ENVIRONMENT 'ESCDELAY' TO '25'
107 *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108 *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109 ACCEPT WS_NUMR FROM LINES
110 ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111 PERFORM 007-OPEN-FILES
112 PERFORM UNTIL E-EXIT
113 MOVE "MENU" TO WS-OP
114 MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115 MOVE SPACES TO WS-CHOICE
116 DISPLAY SS-CLS
117 ACCEPT SS-MENU
118
119 EVALUATE TRUE
120 WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
step

```

A red **!** quotation mark appears on the line **!117**.


This means that application is running and a user action is required at application window.



```

COBGDB GnuCOBOL GDB Interpreter
100 10 COLUMN PLUS 2 TO WS-ERROR.
101
102 PROCEDURE DIVISION.
103 001-START.
104 SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105 SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106 SET ENVIRONMENT 'ESCDELAY' TO '25'
107 *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108 *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109 ACCEPT WS_NUMR FROM LINES
110 ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111 PERFORM 007-OPEN-FILES
112 PERFORM UNTIL E-EXIT
113 MOVE "MENU" TO WS-OP
114 MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115 MOVE SPACES TO WS-CHOICE
116 DISPLAY SS-CLS
117 !ACCEPT SS-MENU
118
119 EVALUATE TRUE
120 WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Running

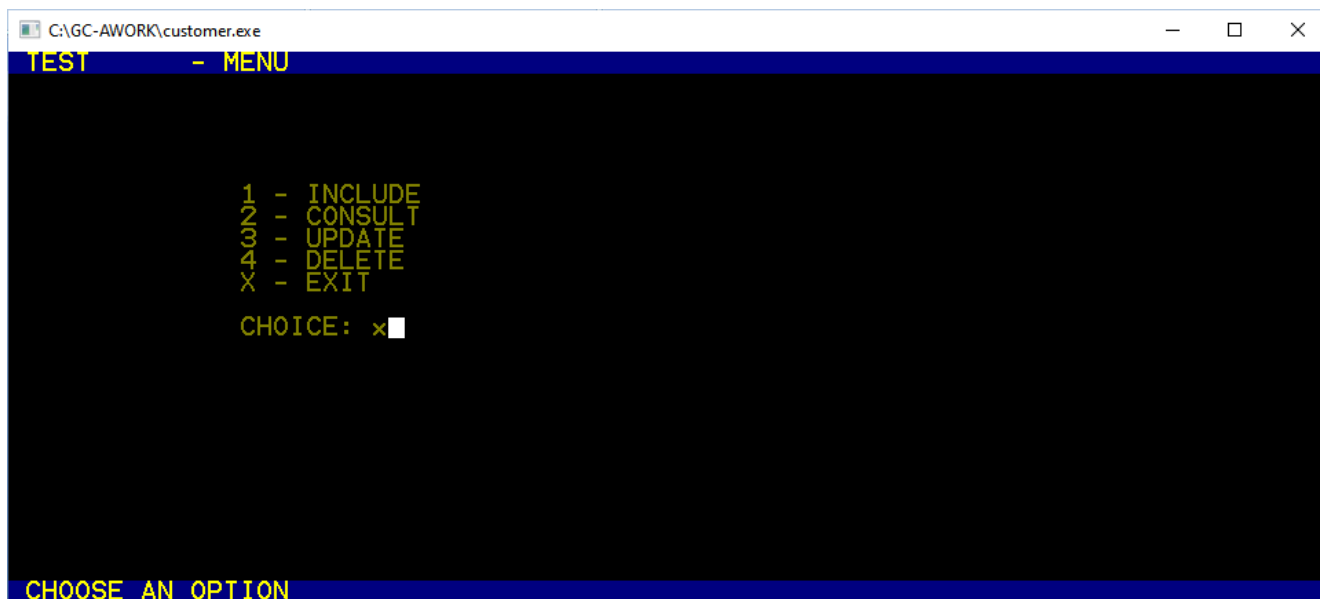
```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	22	

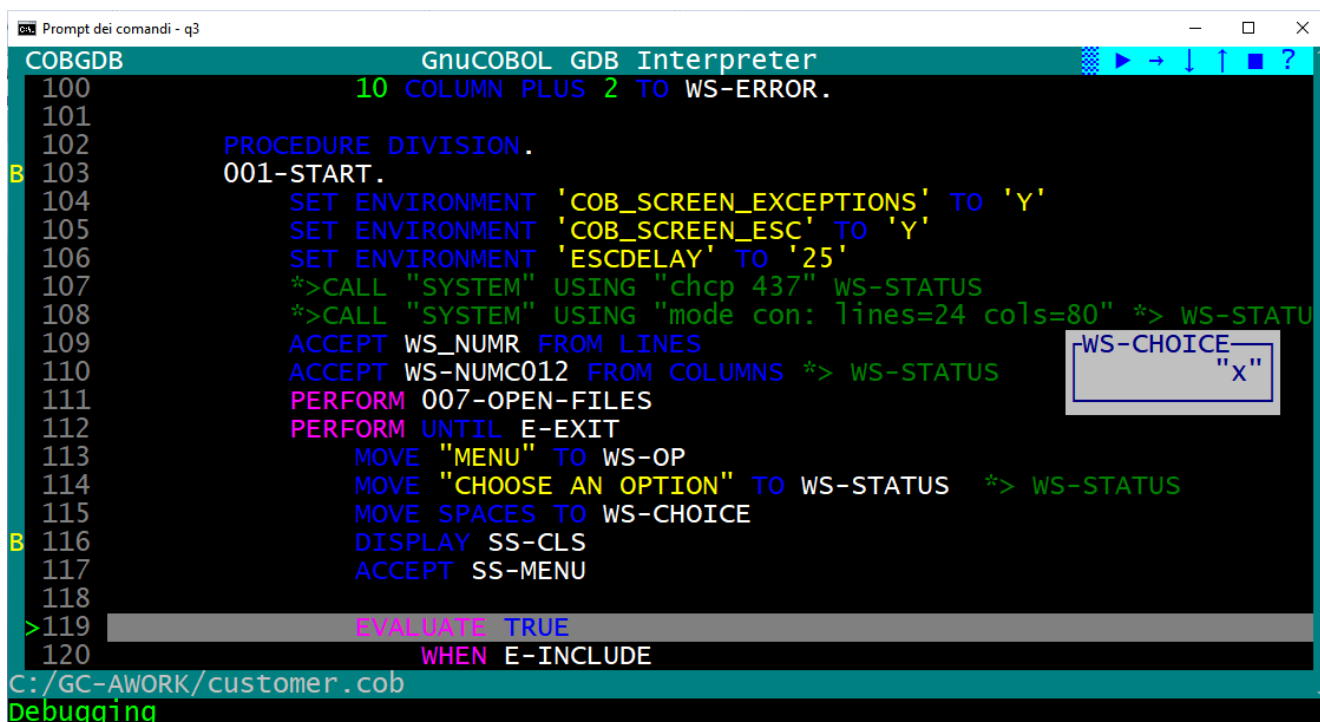
The application is running in a separate window.


The 'Accept' command switches the focus to application windows.

After a user action on application screen (ex type the "X" choice and Enter) it is necessary to click again on the 'debugger' window to continue debugging.



go back to debugger window: the ACCEPT statement has been executed:



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	23	

2.8. Pop-up Variable windows

During a debugging session COBGDB shows variable content.

Blue frame and values : variables of executing cobol statement

Black frame and values : variables of last executed cobol statement.

Sample:



```


COBGDB GnuCOBOL GDB Interpreter
262 continue after 0.5 seconds
263
264 continue.
265 DisplayCoverEx. exit.
266
267 TetraminoDisplay.
268   perform varying wRow from 1 by 1 until wRow > 4
269   perform varying wCol from 1 by 1 until wCol > 4
270     compute wLinD = wBaseLin + wRow
271     compute wColD = wBaseCol + wCol
272     if wBlockEle (wRow wCol) = 'X'
B> 273       display wChar at line wLinD column wColD :BC0: blue :FC0: wCol
274     end-if
275   end-perform
276 end-perform
277 continue.
278 TetraminoDisplayEx. exit.
279
280 TetraminoSet
281   move spa
282   evaluate

```

Variables displayed:

- wColor: 1
- wRow: 03
- wCol: 3
- wColD: 41
- wLinD: 9
- wChar: X
- wBlock: X X X X

File: C:/GC-AWORK/GC99TETRIS.COB

DOCUMENT CODE	MODULE: xxxxxxxxxx	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	24	

2.9. File Command

To show this command we use following sample:

cobgdb sample.cbl subsample.cbl subsubsample.cbl -x -lpcurses

where sample.cbl is the main program; it calls

--> subsample.cbl; it calls

--> subsubsample.cbl

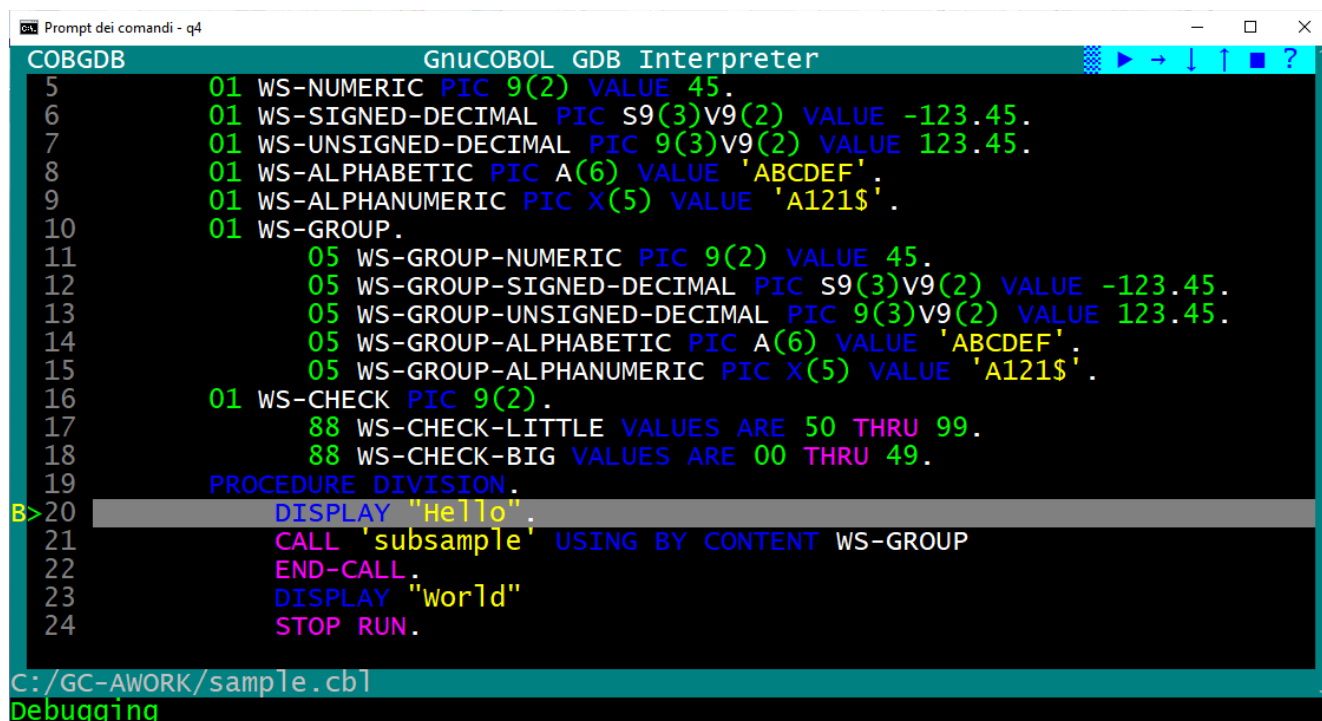
Source code is at <https://github.com/marcsosduma/cobgdb/tree/main/resources>.

This will create a single sample.exe executable.

This example shows that when you need to debug only subsample.cbl or only subsubsample.cbl you need to execute COBGDB with all three programs.

COBGDB sets the B breakpoint at first executable statement of first program "sample.cbl".

here use the R Run command to start the debugging session.




```

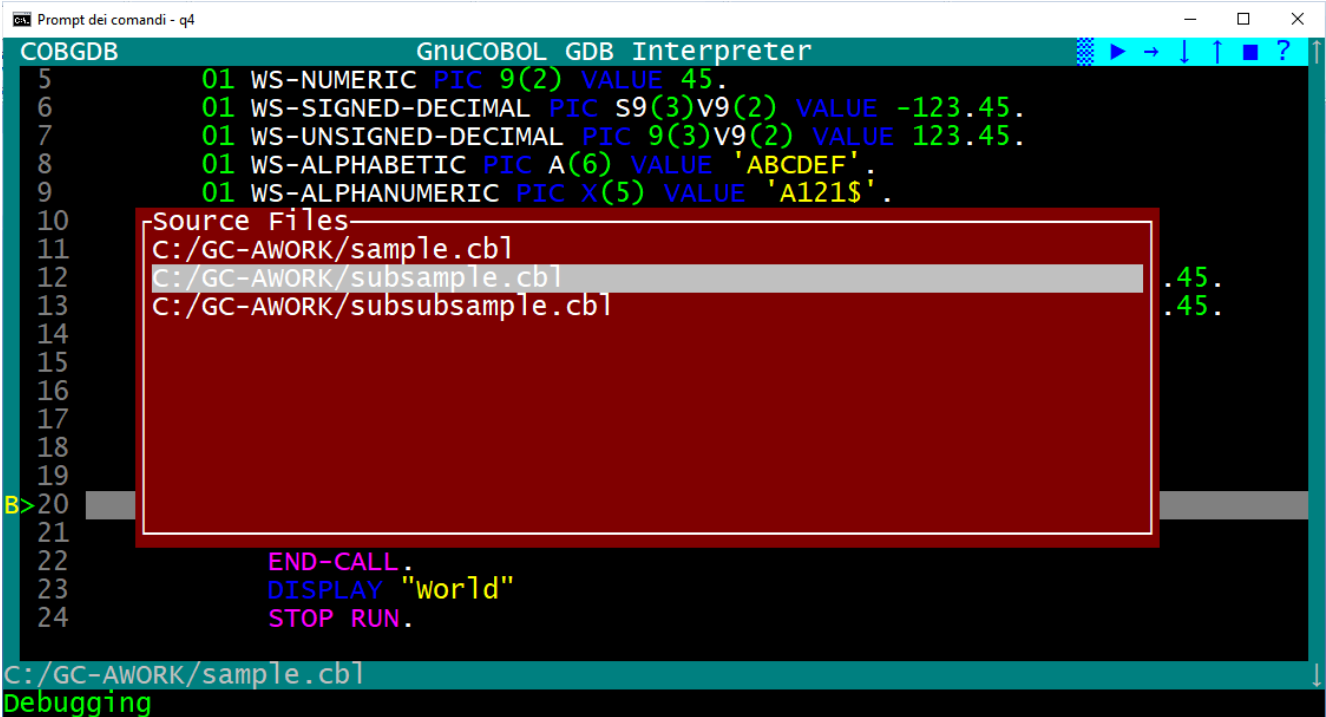
Prompt dei comandi - q4
COBGDB GnuCOBOL GDB Interpreter
5 01 WS-NUMERIC PIC 9(2) VALUE 45.
6 01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
7 01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
8 01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
9 01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
10 01 WS-GROUP.
11     05 WS-GROUP-NUMERIC PIC 9(2) VALUE 45.
12     05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
13     05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
14     05 WS-GROUP-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
15     05 WS-GROUP-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
16 01 WS-CHECK PIC 9(2).
17     88 WS-CHECK-LITTLE VALUES ARE 50 THRU 99.
18     88 WS-CHECK-BIG VALUES ARE 00 THRU 49.
19 PROCEDURE DIVISION.
B> 20 DISPLAY "Hello".
21 CALL 'subsample' USING BY CONTENT WS-GROUP
22 END-CALL.
23 DISPLAY "world"
24 STOP RUN.

C:/GC-AWORK/sample.cbl
Debugging

```


DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	25	

Now you can type the **F File** command and you will have the "Source Files" window.
 In this sample we select the second program in the list (subsample.cbl) and type Enter.

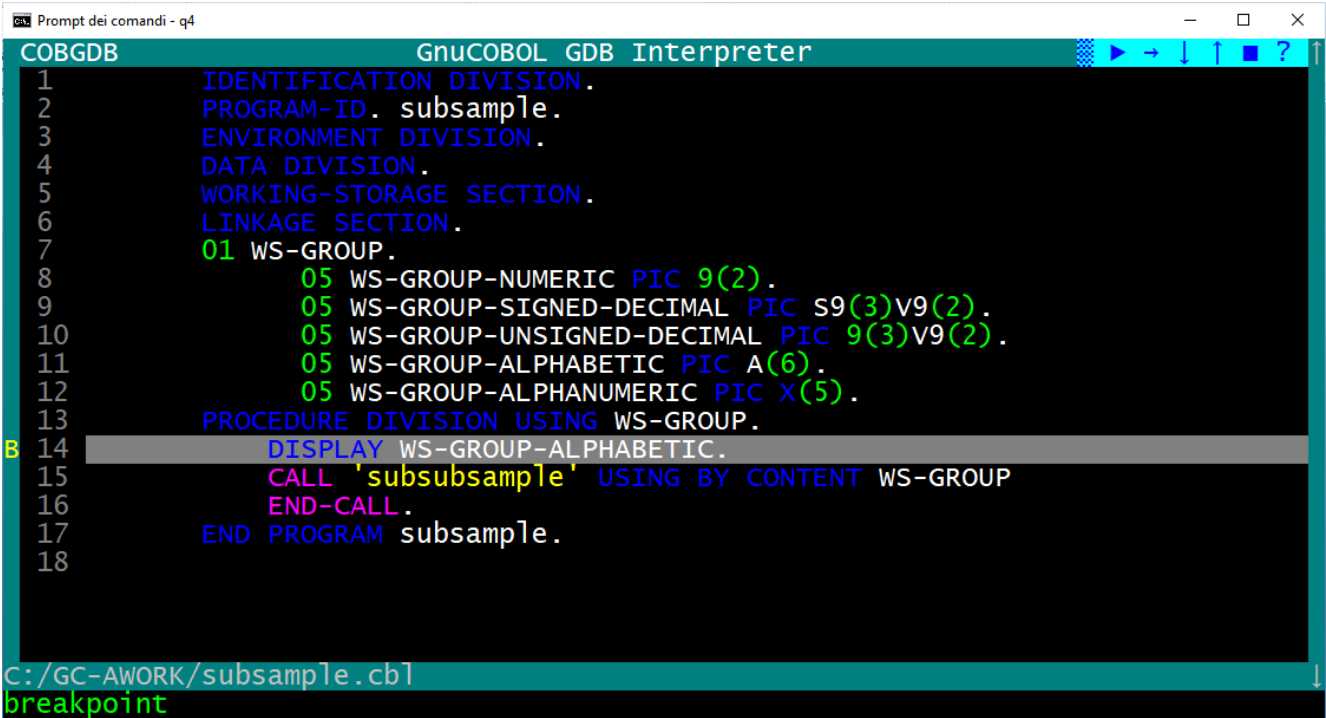


The screenshot shows the COBGDB GnuCOBOL GDB Interpreter window. A red box highlights the 'Source Files' list, which contains three files: C:/GC-AWORK/sample.cbl, C:/GC-AWORK/subsample.cbl (selected), and C:/GC-AWORK/subsubsample.cbl. The main window displays the source code of the selected file, subsample.cbl, with line numbers 5 through 24. The code includes several COBOL data declarations and a procedure division. A 'B' command is entered at line 20, and the status bar at the bottom indicates 'Debugging'.

```

5      01 WS-NUMERIC PIC 9(2) VALUE 45.
6      01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
7      01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
8      01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
9      01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
10
11
12
13
14
15
16
17
18
19
20      B>
21
22      END-CALL.
23      DISPLAY "world"
24      STOP RUN.
  
```


COBGDB shows the selected program source code where in this sample we type a B command at line 14.



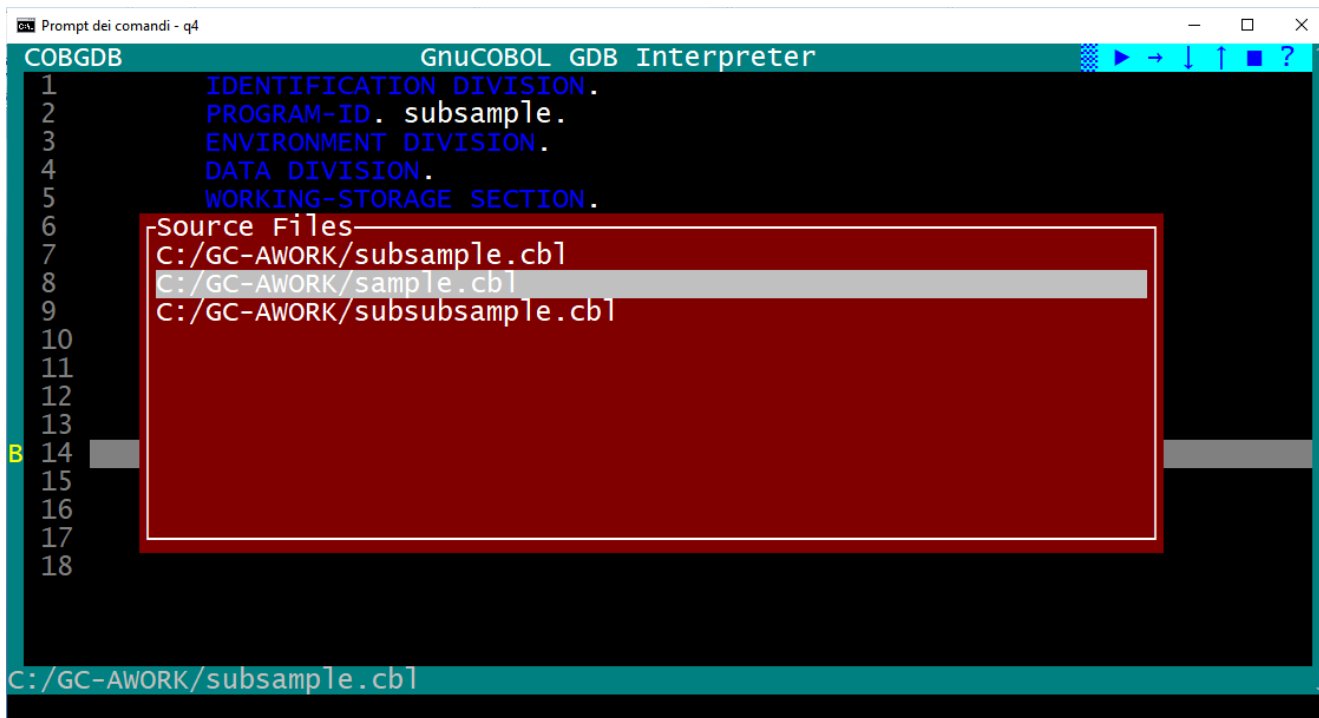
The screenshot shows the COBGDB GnuCOBOL GDB Interpreter window with the source code of subsample.cbl. The code is displayed with line numbers 1 through 18. A 'B' command is entered at line 14, and the status bar at the bottom indicates 'breakpoint'.

```

1      IDENTIFICATION DIVISION.
2      PROGRAM-ID. subsample.
3      ENVIRONMENT DIVISION.
4      DATA DIVISION.
5      WORKING-STORAGE SECTION.
6      LINKAGE SECTION.
7      01 WS-GROUP.
8          05 WS-GROUP-NUMERIC PIC 9(2).
9          05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2).
10         05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2).
11         05 WS-GROUP-ALPHABETIC PIC A(6).
12         05 WS-GROUP-ALPHANUMERIC PIC X(5).
13      PROCEDURE DIVISION USING WS-GROUP.
14      B      DISPLAY WS-GROUP-ALPHABETIC.
15      CALL 'subsubsample' USING BY CONTENT WS-GROUP
16      END-CALL.
17      END PROGRAM subsample.
18
  
```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	26	

now we type the F command again, then select the "sample.cbl" program and press Enter

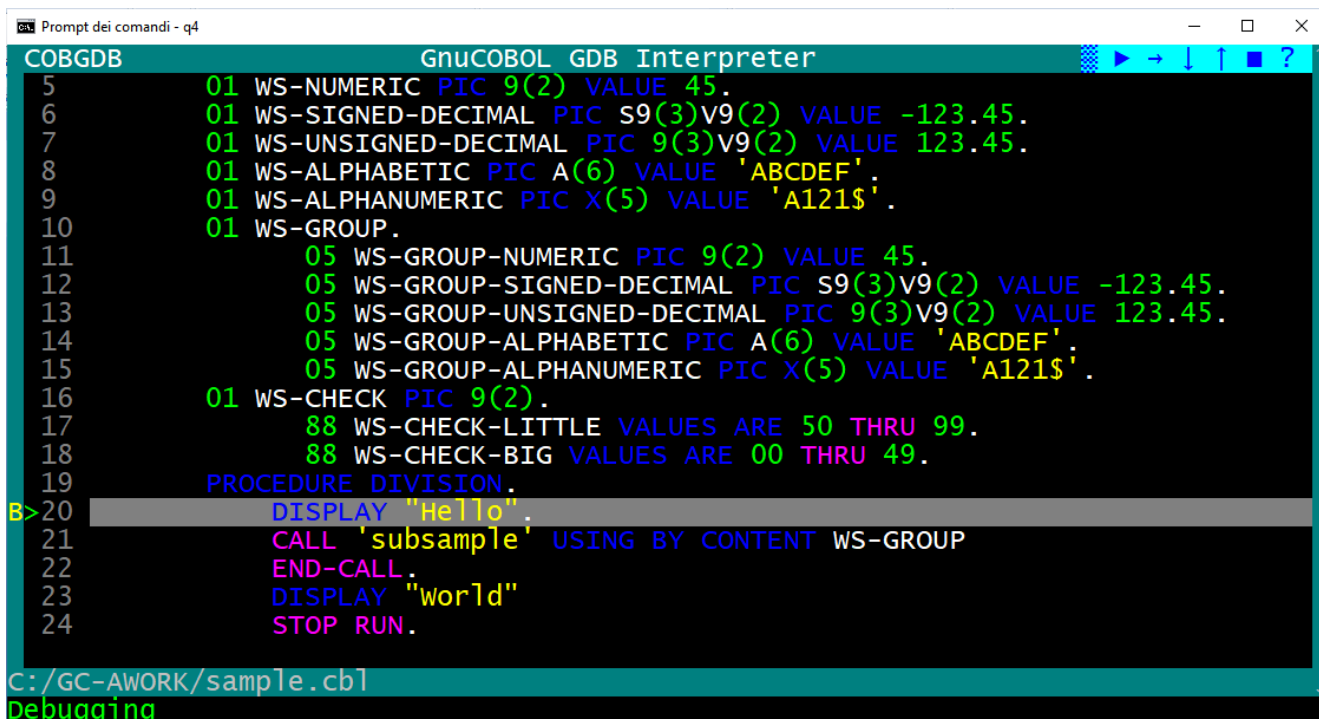


```

COBGDB GnuCOBOL GDB Interpreter
1 IDENTIFICATION DIVISION.
2 PROGRAM-ID. subsample.
3 ENVIRONMENT DIVISION.
4 DATA DIVISION.
5 WORKING-STORAGE SECTION.
6
7 Source Files
8 C:/GC-AWORK/subsample.cbl
9 C:/GC-AWORK/sample.cbl
10 C:/GC-AWORK/subsubsample.cbl
11
12
13
14
15
16
17
18
C:/GC-AWORK/subsample.cbl

```


now we are back to the sample.cbl program to continue the debugging session as we need.



```

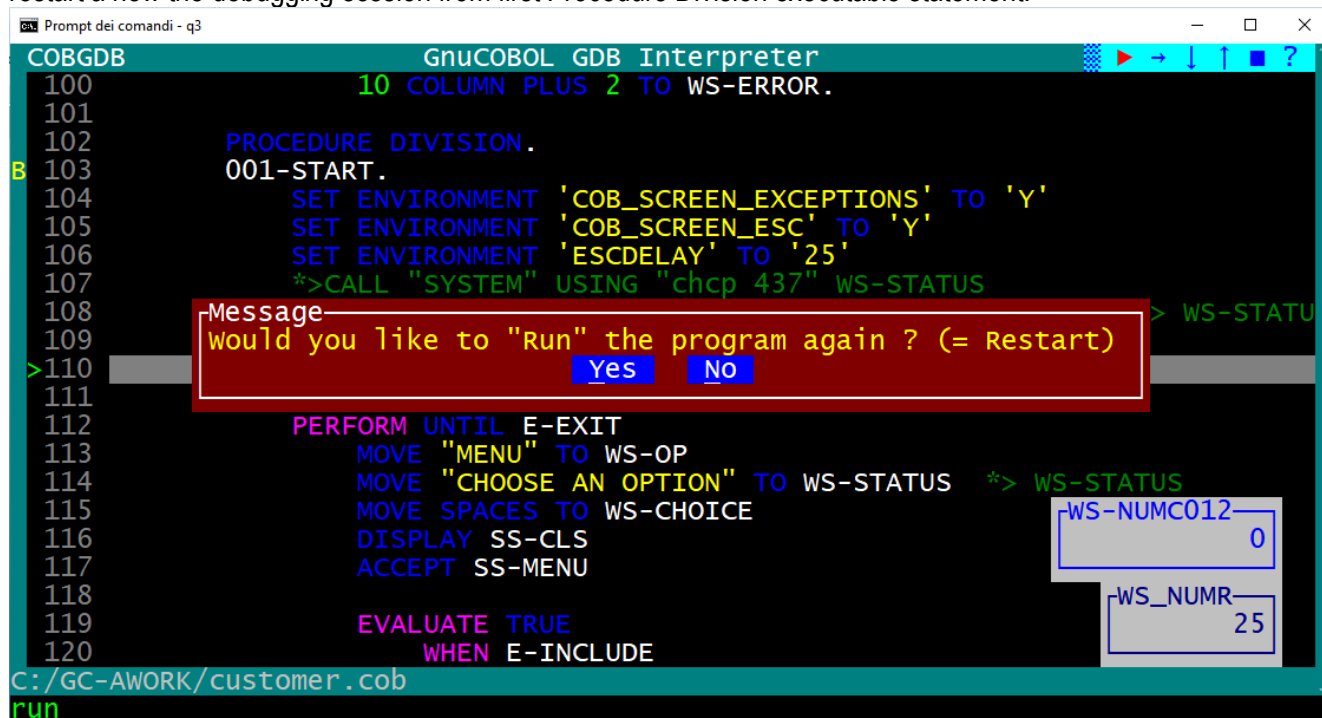
COBGDB GnuCOBOL GDB Interpreter
5 01 WS-NUMERIC PIC 9(2) VALUE 45.
6 01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
7 01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
8 01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
9 01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
10 01 WS-GROUP.
11     05 WS-GROUP-NUMERIC PIC 9(2) VALUE 45.
12     05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
13     05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
14     05 WS-GROUP-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
15     05 WS-GROUP-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
16 01 WS-CHECK PIC 9(2).
17     88 WS-CHECK-LITTLE VALUES ARE 50 THRU 99.
18     88 WS-CHECK-BIG VALUES ARE 00 THRU 49.
19 PROCEDURE DIVISION.
20 DISPLAY "Hello".
21 CALL "subsample" USING BY CONTENT WS-GROUP
22 END-CALL.
23 DISPLAY "world"
24 STOP RUN.
C:/GC-AWORK/sample.cbl
Debugging

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	27	

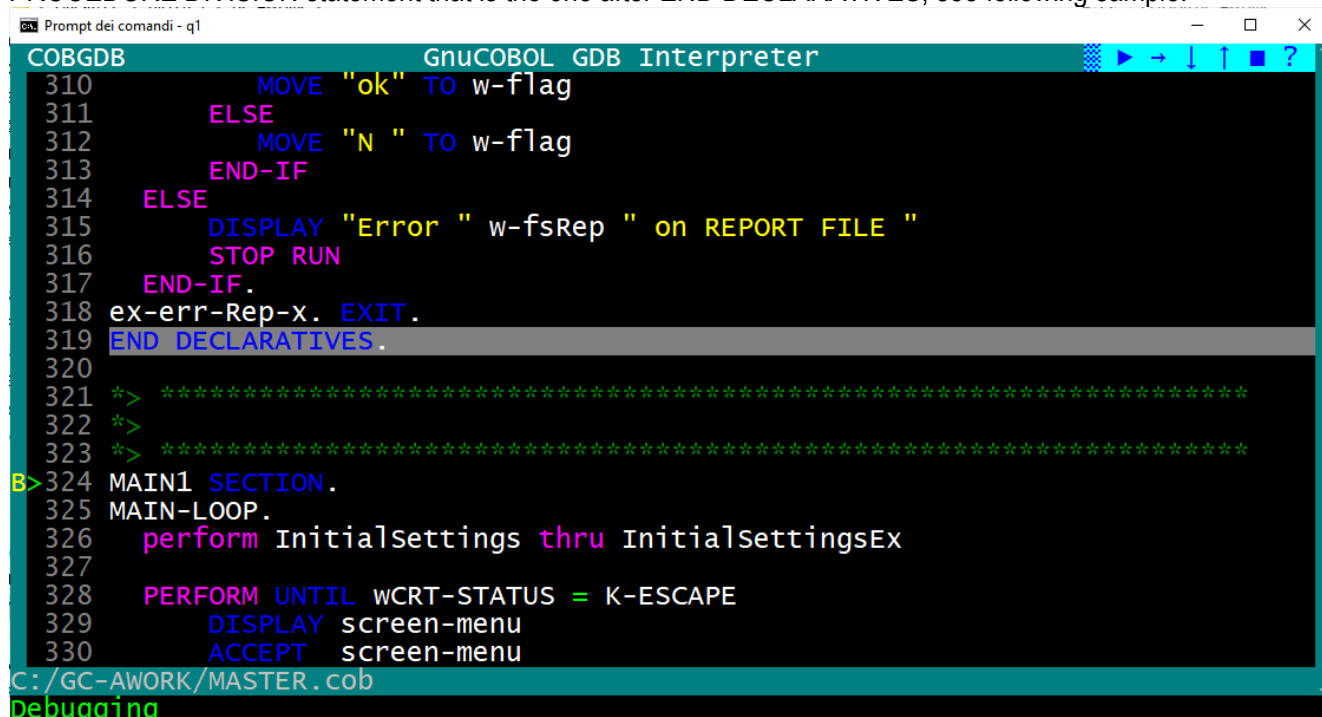
2.10. Run Command

If you click the Run command during a debug session you will receive a confirmation request, because Yes will restart a new the debugging session from first Procedure Division executable statement:




The screenshot shows the GnuCOBOL GDB Interpreter window. The code being debugged is in the COBOL file 'customer.cob'. The program is at line 109, which is a comment line. A red dialog box is displayed in the center of the window, asking: "would you like to 'Run' the program again ? (= Restart)". The dialog has two buttons: "Yes" and "No". The "Yes" button is highlighted. The code in the background includes a PROCEDURE DIVISION with a 001-START section and a main loop section. The status bar at the bottom shows the file path 'C:/GC-AWORK/customer.cob' and the command 'run'.

Note: if program has DECLARATIVES then the first automatic B Breakpoint will be settled at first executable PROCEDURE DIVISION statement that is the one after END DECLARATIVES, see following sample:

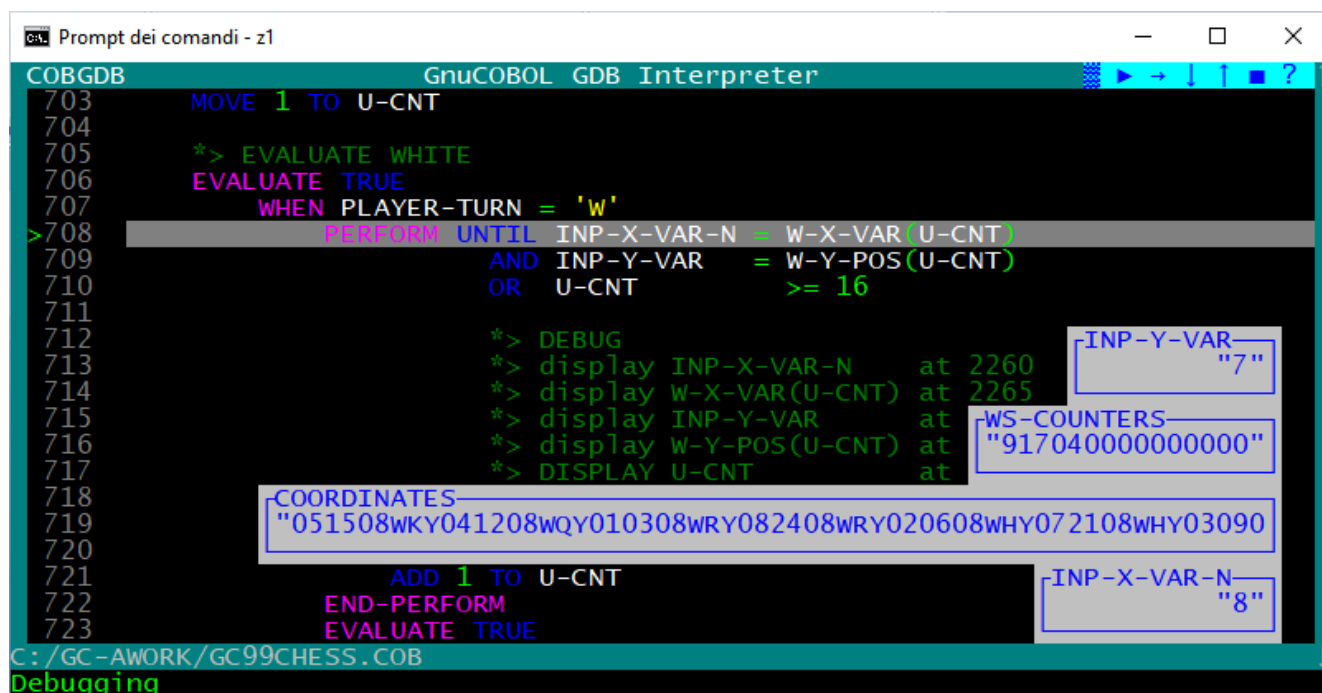


The screenshot shows the GnuCOBOL GDB Interpreter window. The code being debugged is in the COBOL file 'MASTER.cob'. The program is at line 324, which is the start of the MAIN1 SECTION. A breakpoint (B) is set at line 324. The code in the background includes a PROCEDURE DIVISION with a MAIN-LOOP section. The status bar at the bottom shows the file path 'C:/GC-AWORK/MASTER.cob' and the command 'Debugging'.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	28	

2.11. Window Size command

When you start a debugging session, the screen size is 24 x 80 columns.

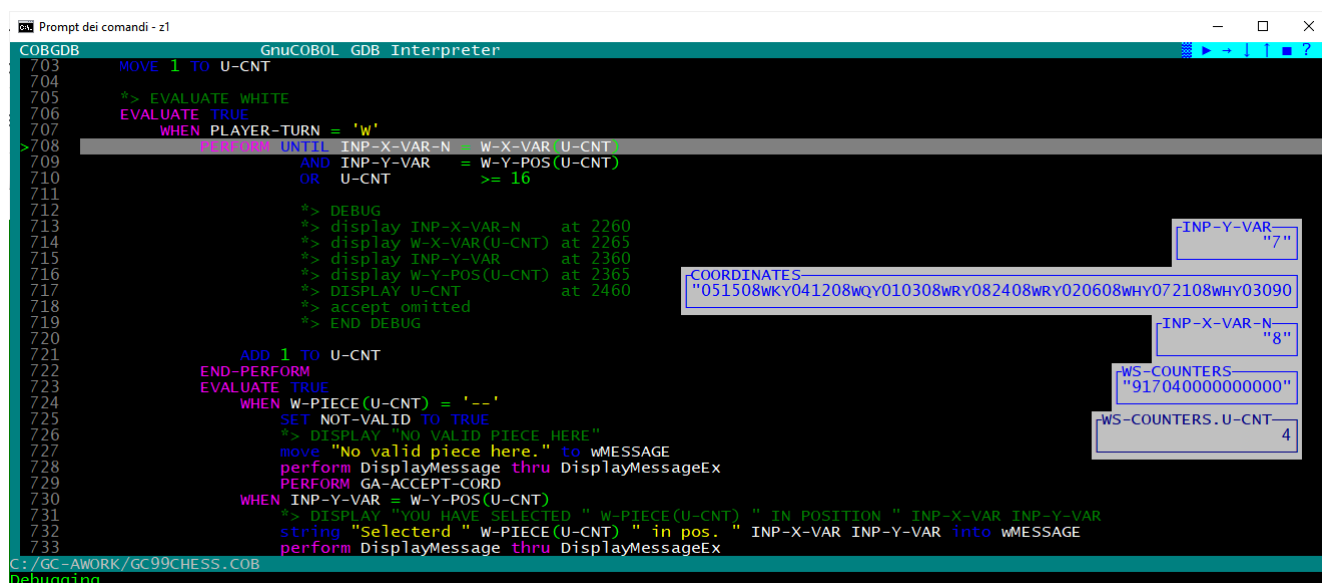


```

COBGDB GnuCOBOL GDB Interpreter
703 MOVE 1 TO U-CNT
704
705 *> EVALUATE WHITE
706 EVALUATE TRUE
707 WHEN PLAYER-TURN = 'W'
>708 PERFORM UNTIL INP-X-VAR-N = W-X-VAR(U-CNT)
709 AND INP-Y-VAR = W-Y-POS(U-CNT)
710 OR U-CNT >= 16
711
712 *> DEBUG
713 *> display INP-X-VAR-N at 2260
714 *> display W-X-VAR(U-CNT) at 2265
715 *> display INP-Y-VAR at
716 *> display W-Y-POS(U-CNT) at
717 *> DISPLAY U-CNT at
718
719 COORDINATES
720 "051508WKY041208WQY010308WRY082408WRY020608WHY072108WHY03090
721
722 ADD 1 TO U-CNT
723 END-PERFORM
724 EVALUATE TRUE
725
C:/GC-AWORK/GC99CHESS.COB
Debugging

```

Type **W** (Window Size) command to switch between two window size of the debugger : 24 x 80 or 34 x 132.




```

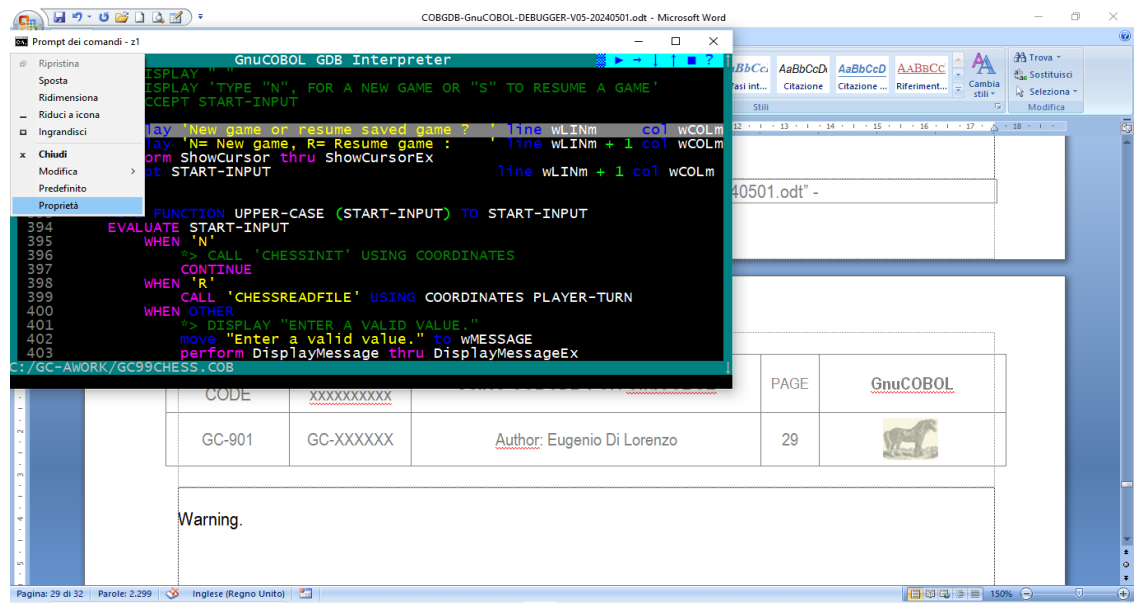
COBGDB GnuCOBOL GDB Interpreter
703 MOVE 1 TO U-CNT
704
705 *> EVALUATE WHITE
706 EVALUATE TRUE
707 WHEN PLAYER-TURN = 'W'
>708 PERFORM UNTIL INP-X-VAR-N = W-X-VAR(U-CNT)
709 AND INP-Y-VAR = W-Y-POS(U-CNT)
710 OR U-CNT >= 16
711
712 *> DEBUG
713 *> display INP-X-VAR-N at 2260
714 *> display W-X-VAR(U-CNT) at 2265
715 *> display INP-Y-VAR at 2360
716 *> display W-Y-POS(U-CNT) at 2365
717 *> DISPLAY U-CNT at 2460
718 *> accept omitted
719 *> END DEBUG
720
721 ADD 1 TO U-CNT
722 END-PERFORM
723 EVALUATE TRUE
724 WHEN W-PIECE(U-CNT) = '--'
725 SET NOT-VALID TO TRUE
726 *> DISPLAY "NO VALID PIECE HERE"
727 move "No valid piece here." to wMESSAGE
728 perform DisplayMessage thru DisplayMessageEx
729 PERFORM GA-ACCEPT-CORD
730 WHEN INP-Y-VAR = W-Y-POS(U-CNT)
731 *> DISPLAY "YOU HAVE SELECTED " W-PIECE(U-CNT) " IN POSITION " INP-X-VAR INP-Y-VAR
732 string "Selected " W-PIECE(U-CNT) " in pos. " INP-X-VAR INP-Y-VAR into wMESSAGE
733 perform DisplayMessage thru DisplayMessageEx
734
C:/GC-AWORK/GC99CHESS.COB
Debugging

```

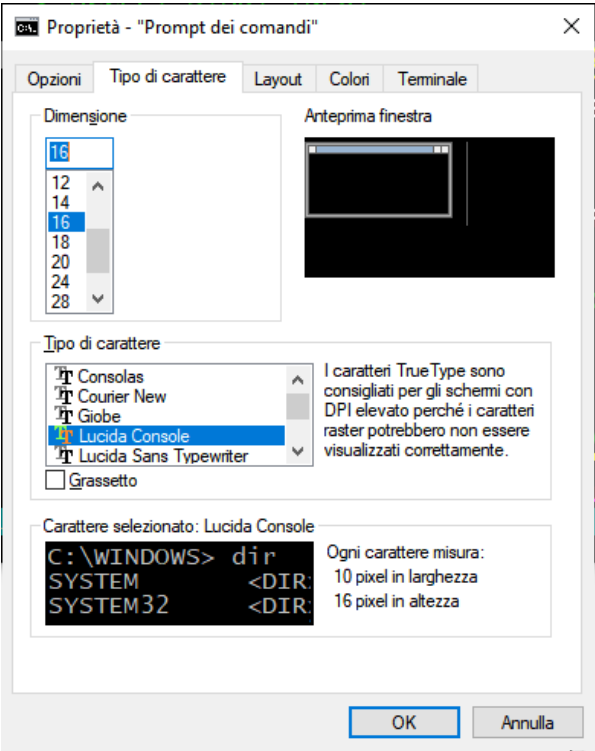
This can be very useful if you write GnuCOBOL code with the new FREE FORMAT source option where each line of code can be longer than the classic 80 characters of FIXED FORMAT.


DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	29	

Warning.
 Make sure the font size is not too large otherwise the W command cannot display the 132 columns.
 For a Windows environment use "Property" menu item:



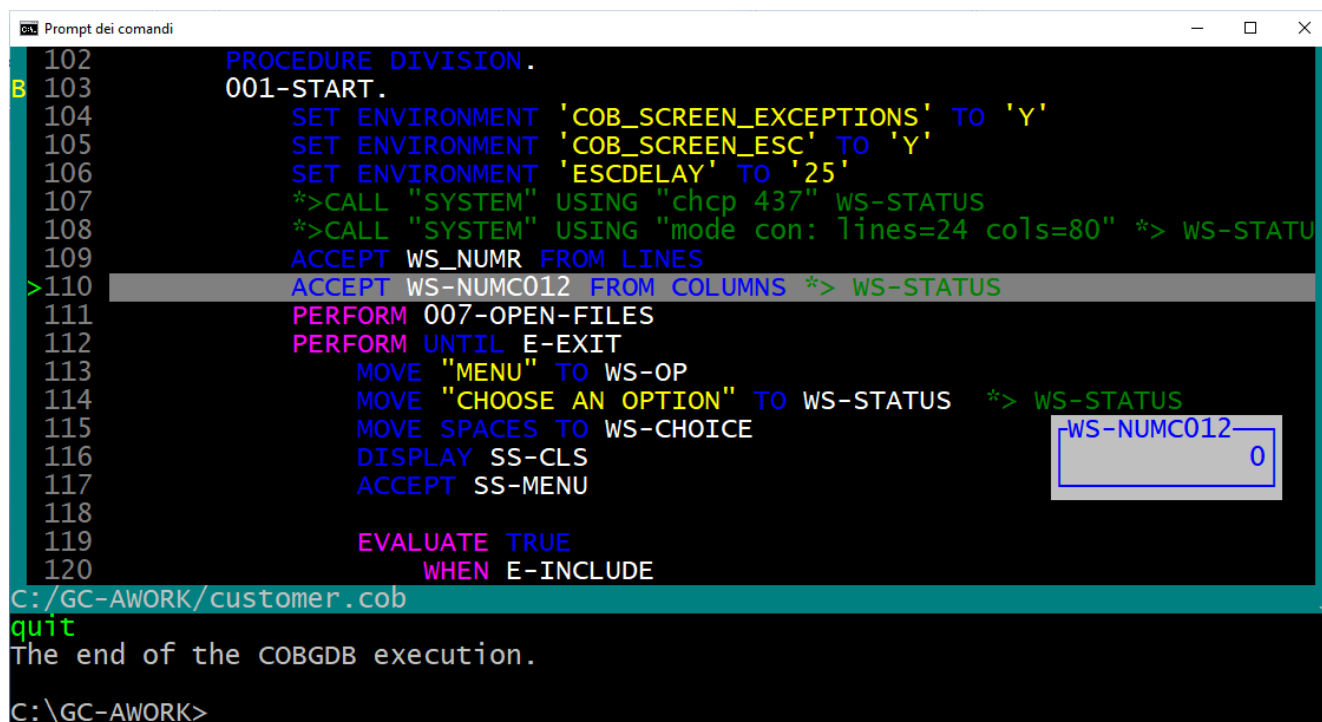
and then select a suitable font, for example Lucida Console of 16 might be adequate.



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	30	

2.12. Quit Command

To close the debug session use the **Q** Quit command or left click with mouse the  button



```

102      PROCEDURE DIVISION.
103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110      > ACCEPT WS-NUMC012 FROM COLUMNS * > WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116              DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
quit
The end of the COBGDB execution.
C:\GC-AWORK>

```

2.13. COBGDB Version


Use command option: `cobgdb --version`

to display COBGDB version informations as follows:

```


COBGDB - GnuCobol GDB Interpreter - version 1.1.0
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
This COBGDB was configured as "MinGW32".
For bug reporting instructions, please see:
<https://github.com/marcsosduma/cobgdb>.
The end of the COBGDB execution.

```

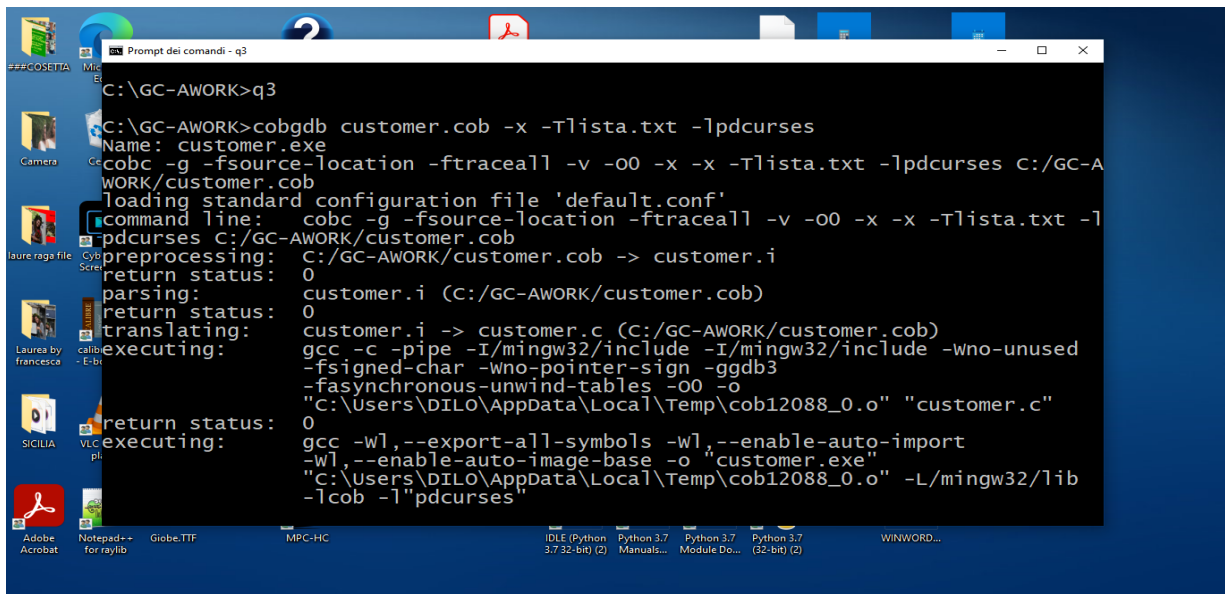
DOCUMENT CODE	MODULE: xxxxxxxxxx	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	31	

3. Document Change Log

CHANGE LOG
<p>Version 1 of 2023.12.12. First release</p> <p>. Version 2 of 2023.12.23. Step by Step sample of use is added Some minor changes</p> <p>. Version 3 of 2024.02.18. Restructured showing new cobgdb screens and features</p> <p>. Version 4 of 2024.04.01 and 20240403. Added EDIT subcommand at H Show Command when viewing the variable from a line of code Added cobgdb --version option</p> <p>. Version 5 of 2024.05.01. Added the W Window size command to change User interface screen size</p>

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	32	

Technical info



```

C:\GC-AWORK>q3
C:\GC-AWORK>cobgdb customer.cob -x -Tlista.txt -lpdcurses
Name: customer.exe
cobc -g -fsource-location -ftraceall -v -o0 -x -x -Tlista.txt -lpdcurses C:/GC-AWORK/customer.cob
loading standard configuration file 'default.conf'
command line: cobc -g -fsource-location -ftraceall -v -o0 -x -x -Tlista.txt -lpdcurses C:/GC-AWORK/customer.cob
preprocessing: C:/GC-AWORK/customer.cob -> customer.i
return status: 0
parsing: customer.i (C:/GC-AWORK/customer.cob)
return status: 0
translating: customer.i -> customer.c (C:/GC-AWORK/customer.cob)
executing: gcc -c -pipe -I/mingw32/include -I/mingw32/include -wno-unused -fsigned-char -wno-pointer-sign -ggdb3 -fasynchronous-unwind-tables -o0 -o "C:\Users\DILO\AppData\Local\Temp\cob12088_0.o" "customer.c"
return status: 0
executing: gcc -w1,--export-all-symbols -w1,--enable-auto-import -w1,--enable-auto-image-base -o "customer.exe" "C:\Users\DILO\AppData\Local\Temp\cob12088_0.o" -L/mingw32/lib -lcob -l"pdcurses"

```